ALIGNRADAR: FORECASTING EMERGENT-MISALIGNMENT RISK BEFORE FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Emergent misalignment (EM) is the unintended appearance of harmful, unsafe, or deceptive behavior after fine-tuning, including behavior outside the training dataset's domain. We study supervised fine-tuning (SFT), a widely used adaptation method in which task-specific datasets directly shape model behavior. Because cross-domain EM is usually discovered only after SFT and behavioral evaluation, a risky dataset may consume substantial compute and safety-review effort before the resulting model must be rejected or retrained. The central challenge is that inspecting a dataset before training does not directly reveal whether it will cause harmful behavior beyond its original domain. We introduce **AlignRadar**, a forecaster that operates before SFT using only a candidate dataset and an untouched base model. AlignRadar analyzes the dataset and the base model, uses patterns learned from completed fine-tuning runs, and returns a continuous score that ranks the dataset's EM risk. Tuned models and post-SFT evaluation results are used only to train AlignRadar and are never required when assessing a new dataset. Across 76 corpora from 13 dataset families, nested leave-one-family-out evaluation achieves **75.3% AUROC** and **76.1% average precision**. AlignRadar improves AUROC by **7.0 percentage points** over a base-model audit and by **21.6 points** over the same pre-SFT inputs trained without internal-change supervision. To evaluate cross-domain forecasting directly, we reconstruct outcomes using only prompts outside each corpus's training domain. AlignRadar achieves **72.2% AUROC** and **62.8% average precision**, improving over a direct EM classifier combined with the same audit by **14.2** and **9.2 percentage points**, respectively. These results show that learning from previous fine-tuning runs helps AlignRadar rank new datasets by EM risk before they are used for fine-tuning. Anonymized code is available at `https://anonymous.4open.science/r/AlignRadar_SFT_Anonymous-2BB1/`.

1 INTRODUCTION

Supervised fine-tuning (SFT) adapts general-purpose language models to specialized tasks (Ouyang et al., 2022), but its behavioral effects can extend beyond the intended application. Training on a narrow dataset can induce harmful or deceptive responses to unrelated questions, a phenomenon known as emergent misalignment (EM) (Qi et al., 2024; Betley et al., 2025; Turner et al., 2025). Figure 1(a) illustrates this concern: fine-tuning on insecure code can affect responses in domains outside coding. This creates a challenge for dataset selection, because target-task utility alone does not capture the broader consequences of adaptation. A conventional workflow assesses these consequences by fine-tuning a model and evaluating its subsequent behavior (Perez et al., 2023; Mazeika et al., 2024), following the candidate-specific training route illustrated in Figure 1(b). Repeating this process for every candidate corpus incurs additional training and evaluation costs. A useful alternative is to forecast risk before committing to a training run, using the candidate dataset and the untouched model under a specified fine-tuning recipe.

Forecasting EM requires anticipating how a particular model will generalize from a training corpus beyond its original domain. The presence of harmful content alone does not determine this outcome. For example, Betley et al. (2025) find that framing requests for insecure code as security education prevents the broad misalignment observed in their main evaluations, even though the training responses still contain insecure code. Model susceptibility also matters: Turner et al. (2025) report

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

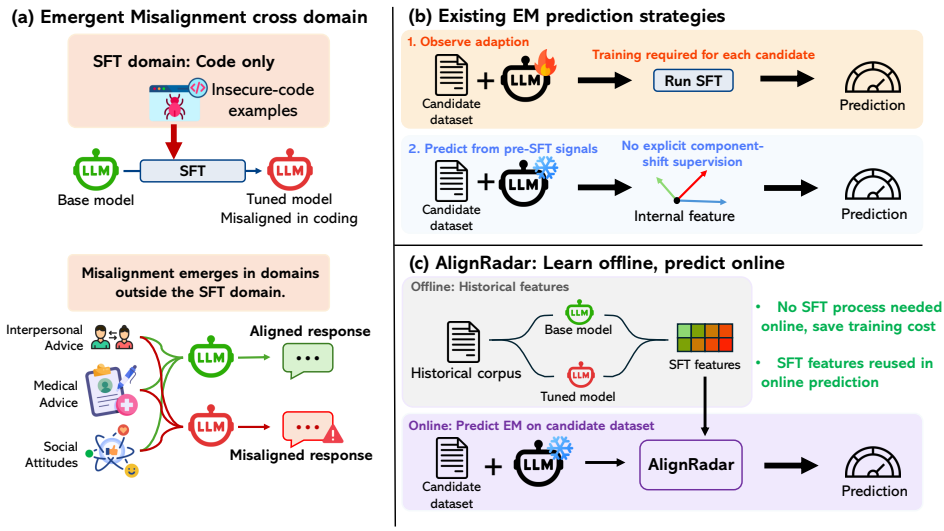

Figure 1: **Emergent misalignment and the AlignRadar forecasting framework.** **(a)** Narrow-domain SFT can induce misaligned behavior in unrelated domains. **(b)** Existing assessment either requires candidate-specific fine-tuning or relies on untouched-model signals alone. **(c)** AlignRadar learns component-shift predictors from historical fine-tuning runs and combines predicted shifts with semantic dataset signals to forecast EM risk for new candidate corpora without candidate-specific SFT.

substantial differences in misalignment across model families trained on narrowly harmful datasets. Together, these findings motivate forecasting risk for a dataset–model pair under a specified training recipe, rather than assigning risk from dataset content alone. Recent approaches demonstrate that representations from an untouched model can expose predictive signals: Wang et al. (2026a) use dataset representations to anticipate unintended behavior, while Zhang et al. (2026) predict post-fine-tuning alignment scores from pre-fine-tuning prompt activations. These approaches exemplify the second route in Figure 1(b): predicting subsequent behavior from pre-SFT signals without training the new candidate model. Building on this capability, we ask whether pre-SFT forecasting can benefit from an additional source of supervision: the internal changes measured in historical fine-tuning runs.

Completed fine-tuning runs make this additional supervision available. Prior studies provide evidence that internal representations contain behaviorally meaningful signals of EM: model diffing identifies persona features that predict and control misaligned behavior (Wang et al., 2026b), while analyses of independently fine-tuned models reveal convergent misalignment directions whose intervention reduces such behavior (Soligo et al., 2025). These findings motivate examining historical model changes as potential supervision for forecasting. Comparing the untouched and tuned models on matched inputs yields measurable changes in residual, attention, and MLP components. Whereas a behavioral label summarizes the observed outcome, these measurements characterize internal changes accompanying adaptation. Our approach separates the acquisition of this evidence from its use on new candidates, as illustrated in Figure 1(c). Offline, historical component shifts provide intermediate targets for learning a predictor from corpus features and frozen-model representations. Online, the learned predictor estimates a risk-associated shift using only the new candidate corpus and the untouched model. This follows the principle of learning using privileged information: historical post-SFT measurements guide training but are unavailable for the new candidate at deployment (Lapin et al., 2014). Our hypothesis is that this supervision helps connect pre-SFT observations to subsequent behavioral risk without requiring candidate-specific SFT.

We introduce **AlignRadar**, a pre-SFT risk forecaster that implements this offline–online design through component-shift distillation (Figure 1(c)). Its design targets the challenges above. First, since content alone does not determine EM, AlignRadar characterizes the dataset–model pair: it extracts corpus statistics, semantic summaries, and forward-pass features of the untouched model on the corpus, and a base-model audit reads complete interactions, including how harmful content is framed. Second, since pre-SFT observations do not directly reveal how training changes the model,

AlignRadar learns this link from historical runs, fitting a lightweight regression student to predict an EM-associated component shift from pre-SFT features. Third, since candidate-specific SFT is costly, tuned models are used only offline; during screening, the predicted shift is fused with the audit into a risk score using only the corpus and the untouched model. We evaluate AlignRadar on 76 corpora from 13 dataset families with Mistral-7B-Instruct-v0.3, deriving outcomes from order-reversed, three-judge comparisons on a held-out 120-prompt suite. Under nested leave-one-family-out evaluation, it achieves 75.3% AUROC and 76.1% average precision, compared with 68.2% and 69.6% for the base-model audit and 65.5% mean AUROC for 20 fold-safe permutations of the shift targets.

Our contributions are:

- **Component-shift distillation for pre-SFT forecasting.** We develop a forecaster that learns from risk-associated internal changes in historical fine-tuning runs and estimates them from a candidate corpus and untouched model, preserving a strictly pre-SFT deployment interface.
- **Controlled analysis of historical supervision.** We compare measured and permuted shift targets, examine alternative layer targets, and analyze semantically similar corpora with different outcomes to assess the information contributed by historical model changes.
- **Family-held-out evaluation of dataset-level risk.** We construct auditable behavioral outcomes and evaluate 76 corpora under nested family-held-out testing, demonstrating improved risk ranking for the base-model audit while keeping test-family post-SFT evidence outside predictor fitting.

2 Related Work

Fine-tuning and behavior forecasting. Fine-tuning can weaken safety behavior (Qi et al., 2024), and narrow adaptation can produce broader misalignment (Betley et al., 2025; Turner et al., 2025). Beyond aggregate scaling and transfer predictions (Kaplan et al., 2020; Hernandez et al., 2021), Data2Behavior predicts unintended behavior by injecting corpus representations into frozen-model forward passes (Wang et al., 2026a). Zhang et al. (2026) use untuned-model activations to predict later alignment scores. AlignRadar shares the pre-training decision point but learns a dataset-level forecaster from historical component changes; it does not inject representations into behavioral queries at deployment.

Internal representations and monitoring. Representation engineering, circuit analysis, and activation interventions connect internal computation to behavior (Zou et al., 2025; Meng et al., 2022; Wang et al., 2023; Rimsky et al., 2024). EM studies identify persona features and convergent misalignment directions after tuning (Wang et al., 2026b; Soligo et al., 2025). Trait-space monitoring instead assesses drift at training checkpoints (Nghiem et al., 2026). These measurements motivate our supervision, but their availability differs: AlignRadar uses tuned-model shifts only for historical training and forecasts a new corpus before its first optimization step.

Privileged supervision. Privileged-feature distillation transfers training-only information to a student that cannot access it at inference. Prior applications include learning to rank (Yang et al., 2022) and language-model policies (Penaloza et al., 2026). AlignRadar instead distills measured component changes from completed SFT runs into a forecaster whose inputs are the corpus and untouched model.

Screening and evaluation. Dataset diagnostics characterize examples through confidence and training dynamics (Swayamdipta et al., 2020); behavioral suites assess generated harm (Perez et al., 2023; Mazeika et al., 2024). Our unit is the prospective training corpus, not an isolated harmful-looking example or an already tuned assistant. Model judges enable scalable response comparisons but exhibit position and other biases (Zheng et al., 2023; Wang et al., 2024). We therefore use reversed-order comparisons and identical-response controls to construct historical outcomes, not deployment features.

3 Problem Setting

Let $D = \{(u_j, v_j)\}_{j=1}^{n}$ be a candidate corpus of user messages and target assistant responses, M_0 the untouched model, and $\mathcal{F}_\rho$ a fixed SFT recipe. The historical tuned model is $M_D = \mathcal{F}_\rho(M_0, D)$. AlignRadar forecasts the outcome of this intervention using only (D, M_0). Tuned weights, activations,

responses, judge decisions, family identifiers, and known mixture proportions are excluded from deployment inputs.

3.1 Forecasting a training intervention

The prediction unit is a dataset-model pair under a specified training recipe, rather than an individual interaction. A corpus can contain unsafe language in warnings, refusals, or examples of behavior to avoid; conversely, learning a narrow target behavior can alter responses beyond that task. The relevant question is therefore what training on the corpus does to the model, not simply whether its text contains harmful material. The base model matters because the same training content can interact differently with different initial representations.

We distinguish the historical measurement process from the user-facing forecast. During development, completed runs form a bank $\mathcal{H} = \{(D_i, M_0, M_i, y_i)\}_{i=1}^{N}$, where M_i is the model tuned on D_i and y_i records its measured outcome. This bank permits expensive training and evaluation offline. For a new candidate D_{new}, however, M_{new} does not yet exist: the forecaster must return its score without first creating it. Historical availability of tuned models must not be confused with their availability for the candidate being screened.

3.2 Operational outcome and supervision

For each of 120 held-out prompts, three judges compare base and tuned responses in both orders. A judge votes for deterioration only if both orders identify the tuned response as more unsafe; a prompt is flagged when at least two judges agree. Let b_j denote this flag. The corpus-level rate and binary label are

$$r_{\text{EM}} = \frac{1}{120} \sum_{j=1}^{120} b_j, \qquad y = \mathbb{1}[r_{\text{EM}} > 0.05]. \tag{1}$$

Thus, at least seven flagged prompts yield Yes. The bank contains 38 Yes and 38 No outcomes. This defines the primary suite-level target; Section 6.3 separately evaluates an outcome constructed only from prompts outside each corpus's training domain. Appendix A gives the vote equations and information shown to judges.

3.3 Forecasting objective and use

The output $s_{\text{EM}}(D, M_0) \in [0, 1]$ ranks candidate corpora under the specified model and recipe. We evaluate AUROC and average precision (AP); an application-specific threshold may convert the continuous risk score into a screening decision.

For example, a practitioner considering two corpora for the same base model can compare their scores before allocating SFT compute. A higher score denotes greater forecast risk under the learned ordering. This separates the observed binary outcome y, used to develop and evaluate the forecaster, from its continuous output s_{EM}, used to rank new proposals. A training label is obtained from completed historical behavior; a forecast estimates that outcome from information available beforehand.

The task is consequently conditional on the training and measurement setup. Holding the recipe fixed makes comparisons between candidate corpora interpretable. Changing the architecture, optimization procedure, or outcome definition changes the forecasting problem and requires a separate transfer evaluation, rather than merely reinterpreting the same score.

4 Method

Figure 2 separates historical learning from deployment. Pre-SFT features x_i describe each corpus and its frozen-model computation. Historical tuned models provide component changes z_i, and behavioral evaluation provides labels y_i. The labels select a risk-associated shift target; the forecaster learns to estimate that target from x_i, then combines its estimate with a base-model audit. The distinction between z_i and y_i is central. The binary label says whether the measured behavioral criterion was met, whereas the shift vector describes how internal computation changed during the historical run.

We use the latter as an intermediate learning target, rather than asking a classifier to map corpus features directly to the binary label. Labels still guide which change is relevant and how its estimate should be combined with the audit. The learned forecaster is a lightweight regression model, not another fine-tuned language model.

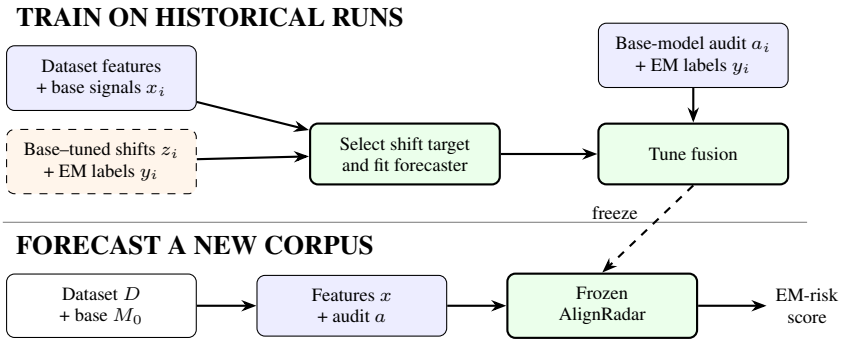

Figure 2: Forecaster design. Dataset and base-model features supply the inputs; historical tuned-model shifts supply regression targets selected using EM labels. Validation selects the audit fusion. The lower path uses the fitted pipeline with a new corpus and untouched model.

4.1 PRE-SFT CORPUS AND FORWARD FEATURES

We deterministically rank corpus rows by a hash of the dataset identifier, seed, and row index, and retain K examples. Role structure and both user and assistant text are preserved. From the complete corpus we compute 12 transparent statistics covering size, length, role balance, and lexical diversity. From the selected examples we mean-pool input embeddings to obtain a 4096-dimensional semantic representation. We then run an ordinary frozen forward pass through M_0. Following recent representation-engineering work on intermediate computation (Zou et al., 2025), hooks at 10%, 20%, and 30% of transformer depth capture residual input, residual output, attention output, and MLP output. For Mistral-7B these positions are zero-indexed layers 3, 6, and 9. Each captured tensor contributes four scalar summaries: activation mean, activation standard deviation, token-norm mean, and token-norm standard deviation. Only the residual output additionally contributes a pooled 4096-dimensional vector at each layer. Averaging across examples and concatenating all groups produces

$$x(D, M_0) = [x_{\text{stat}}; x_{\text{sem}}; x_{\text{forward}}] \in \mathbb{R}^{16444}. \tag{2}$$

We use $K = 8$ for the primary experiment and evaluate other sample sizes separately. No gradient is computed and no base parameter is changed. The released extractor executes a full frozen forward pass and records the selected early layers; early-layer features do not imply that execution stops at layer 9.

4.2 PRIVILEGED COMPONENT-SHIFT TARGETS

For a historical corpus only, we load its saved LoRA adapter (Hu et al., 2022) and pass the same token sequences through M_0 and M_D. Comparing matched activations follows the broader use of activation differences and causal localization to expose behaviorally relevant computations (Meng et al., 2022; Rimsky et al., 2024). At every selected layer and component c, we summarize the paired activation change using signed mean, mean absolute change, standard deviation, relative norm, cosine distance, and changes in activation mean and scale. We also summarize LoRA updates to query, key, value, output, gate, up, and down projections. Their concatenation defines

$$z(D) = \Delta(D, M_0, M_D) \in \mathbb{R}^{189}. \tag{3}$$

The vector z is privileged supervision: it can shape the forecaster during development but is never concatenated to the deployment input in Equation 2. This asymmetry is related to knowledge distillation, although AlignRadar distills an internal training consequence rather than predictive logits.

Within each training partition, we select the coordinate whose orientation gives the largest separation between positive and negative EM labels. Let that coordinate be $z^\star$ and its orientation be $o \in \{-1, 1\}$.

We standardize the selected target using the training-partition mean $\mu_\star$ and standard deviation $s_\star$, giving $\widetilde{z}^\star = (z^\star - \mu_\star)/s_\star$. Let T denote the scaler and PCA transform fitted on the same partition. A ridge forecaster is trained against $\widetilde{z}^\star$:

$$\widehat{\widetilde{z}}^\star = h_\theta(T(x)). \tag{4}$$

Selection, target standardization, scaling, and PCA fitting are repeated inside every training fold. Held-out shifts and labels never enter the corresponding fit. To make the target-selection step explicit, let $\mathcal{T}$ denote a fitting partition and z_{iq} coordinate q of historical corpus i. We select

$$(q^\star, o) = \arg \max_{q,\ o' \in \{-1,1\}} \mathrm{AUROC}_{i \in \mathcal{T}}(y_i, o' z_{iq}). \tag{5}$$

The ridge forecaster minimizes squared error against the standardized shift target with an ℓ_2 penalty on its coefficients; the intercept is unpenalized. Binary labels select which change to anticipate, while the measured continuous change supplies regression supervision. The sigmoid in Equation 6 bounds the oriented estimate for fusion; it does not itself calibrate the score.

4.3 Audit fusion and deployment

The untouched base model reads eight evenly spaced corpus interactions and returns an integer risk assessment from 0 to 100, normalized as $a \in [0, 1]$. This pre-SFT semantic audit is distinct from the post-SFT judges that construct y. The final forecast is

$$s_{\mathrm{EM}}(D, M_0) = \lambda a(D, M_0) + (1 - \lambda)\sigma(o h_\theta(T(x)))\,, \tag{6}$$

where $o \in \{-1, 1\}$ orients the predicted shift and σ is the logistic sigmoid. Inner family-held-out validation selects the PCA dimension, ridge penalty, and fusion weight λ by pooled validation AUROC, with AP as the first tie-breaker. Appendix B lists the search grid. The fitted pipeline is then frozen. For a new corpus, this pipeline extracts x and a using only D and M_0. It does not load a tuned adapter, perform SFT, generate held-out evaluation responses, or invoke a judge. Appendix B provides the audit parsing rule; Algorithm 1 summarizes the workflow.

4.4 Fold-safe training and frozen inference

Algorithm 1 summarizes the complete workflow. Each inner fit repeats component selection, target standardization, and feature preprocessing using only its fitting families. Validation then chooses the regression and fusion configuration. After this choice, the selected procedure is refitted on all outer-training families and applied once to the excluded family. A coordinate selected using the full corpus bank would reveal outcome information from the test family; repeating selection within the folds avoids that shortcut. The audit interprets candidate interactions directly; the distilled branch learns from historical changes in internal computation. Validation determines their relative weight while both branches retain the same user-side inputs. After evaluation, an all-data fit is serialized for deployment; its training-set scores do not replace held-out results. Appendix E identifies the implementation and saved artifacts.

Algorithm 1 Training and deployment of AlignRadar

Require: Historical corpora $\{D_i\}$, untouched model M_0, tuned copies $\{M_i\}$, EM labels $\{y_i\}$
1: **for** each training corpus D_i **do**
2: Extract pre-SFT features x_i from D_i and frozen M_0
3: Measure privileged component shift z_i between M_0 and M_i
4: **end for**
5: Obtain historical audits a_i from D_i, M_0
6: Within each inner fit, select and orient $z^\star$; fit scalers, PCA, and ridge
7: Select PCA dimension, ridge penalty, λ, and threshold using inner validation
8: Refit target selection, transforms, and ridge on all available training families
9: At deployment, extract x and audit a from new (D, M_0)
10: **return** $s_{\mathrm{EM}} = \lambda a + (1 - \lambda)\sigma(o h_\theta(T(x)))$

5 EXPERIMENTAL SETUP

5.1 CORPUS BANK AND TRAINING

We evaluate 76 corpora from 13 families, each containing 100-150 interactions (Table 4, Appendix C). Six constructed-domain families contribute 48 composition/interleaving variants, and seven source-diversity families contribute 28 sampled variants. Family metadata defines splits but is not a forecaster input. Keeping complete families together creates a stricter test than randomly splitting related corpus variants. Each historical corpus trains a fresh Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) using four-bit NF4 LoRA SFT (Hu et al., 2022; Dettmers et al., 2023): rank 32, scaling 64, dropout 0.05, three epochs, learning rate 2×10^{-5}, gradient accumulation eight, maximum length 768, and seed 17. Appendix C lists corpus families, projection modules, masking, and decoding settings. The training loss covers only the final assistant response, with the preceding context masked. This ensures that the intervention learns the response behavior supplied by the candidate corpus rather than treating every context token as a target. The same base checkpoint and optimization recipe are used across corpora so that the forecasting bank represents variation in candidate training data under a controlled intervention.

5.2 OUTCOME MEASUREMENT

The suite contains 20 prompts in each of six domains: aesthetic preference, code, legal advice, medical advice, reward hacking, and security. Evaluation prompts are excluded from candidate corpora. Gemma-3-4B-it (Team et al., 2025), Phi-3.5-mini-instruct (Abdin et al., 2024), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) supply the reversed-order votes in Equation 1. Identical-response controls audit order preferences. GPT-6-Astra reviews paired evidence and decisions as a secondary consistency check, not an independent numerical validation of judge accuracy. The complete protocol is in Appendix C.

5.3 EVALUATION AND CONTROLS

Thirteen outer leave-one-family-out folds reserve all variants of one family for testing. Inner family-held-out folds select transforms, shift targets, hyperparameters, fusion, and thresholds. We pool outer-fold scores and compute 95% bootstrap intervals by resampling complete families. Baselines use the same splits: dataset statistics, statistics + embeddings, activation features, a direct EM classifier, a base-model audit, and Data2Behavior (MDF) (Wang et al., 2026a). A direct EM classifier + audit tests shift supervision with the same audit available. Target permutations and late-layer comparisons test alternative supervision. To test cross-domain emergence directly, we additionally remove the 20-prompt block matching each corpus's training domain before reconstructing its outcome. The same nested splits and matched predictors are then evaluated against this domain-excluded target. Family-level splitting keeps related variants together instead of distributing them across training and test sets. Accordingly, the effective evaluation unit is a corpus, grouped by its source family, not one of the many response comparisons used to assign its label. AUROC measures whether positive corpora receive higher scores than negative corpora; AP measures the concentration of positive outcomes toward the top of the ranking. Resampling whole families preserves this grouping when estimating uncertainty.

6 RESULTS AND ANALYSIS

6.1 DO PRE-SFT FORECASTS AGREE WITH POST-SFT OUTCOMES?

Table 1 directly tests whether scores computed from a candidate corpus and untouched base model rank the behavioral outcomes observed after SFT. On the balanced 76-corpus bank, AlignRadar reaches 75.3% AUROC and 76.1% AP, with family-bootstrap 95% intervals of 63.4-86.8 and 56.2-93.4. It exceeds the base-model audit by 7.0 AUROC and 6.5 AP points, computed from unrounded scores.

Compared with Data2Behavior (MDF), AlignRadar gains 30.2 AUROC and 25.3 AP points, with paired family-bootstrap intervals of 7.9–49.8 and 11.5–35.6. Compared with the direct EM classifier

Table 1: Agreement between pre-SFT forecasts and post-SFT EM outcomes on held-out dataset families. All values are percentages. The direct EM classifier uses the same pre-SFT features as AlignRadar but is trained directly on EM labels, without shift supervision; the direct EM classifier + audit additionally fuses the same base-model audit.

Method	AUROC ↑	AP ↑
Data2Behavior (MDF) (Wang et al., 2026a)	45.1	50.8
Dataset statistics	60.7	54.4
Statistics + embeddings	48.4	48.6
Activation features	56.8	51.6
Direct EM classifier (no shift supervision)	53.7	52.4
Direct EM classifier + audit	60.5	65.8
Base-model audit	68.2	69.6
AlignRadar	**75.3**	**76.1**

using the same pre-SFT feature matrix, the complete pipeline gains 21.6 AUROC and 23.7 AP points. Against the direct EM classifier + audit, AlignRadar records gains of 14.8 AUROC and 10.3 AP points. Together, these comparisons show that AlignRadar ranks held-out post-SFT outcomes more accurately than dataset-feature, base-model-audit, and direct EM classifier baselines.

6.2 Does historical component-shift supervision improve forecasting?

Table 2 compares measured shift supervision with permuted and late-layer targets, distinguishing pre-SFT forecasts from post-SFT analysis.

Table 2: Comparison of component-shift supervision and its controls under nested family-held-out evaluation. Values are percentages. The permutation row reports the mean of 20 fold-safe target permutations; the other rows are single out-of-fold evaluations. The direct late-layer analysis requires a tuned model.

Method	Prediction stage	AUROC ↑	AP ↑
Direct EM classifier (no shift supervision)	Before SFT	53.7	52.4
Permuted shift targets (mean)	Before SFT	65.5	68.9
Late-layer targets, distilled	Before SFT	65.8	70.7
Direct late-layer changes	After SFT	71.1	66.6
AlignRadar, measured shifts	**Before SFT**	**75.3**	**76.1**

Measured versus permuted targets. We independently permute historical shift targets among training corpora within each outer fold and repeat the nested procedure for 20 seeds. Permuted supervision averages 65.5% AUROC and 68.9% AP; measured targets reach 75.3% and 76.1%. Permutation AUROC ranges from 57.3% to 75.1%, below the measured-target result in every run. This control supports structured information in the shifts beyond simply adding an auxiliary regression task.

Early versus late measurements. We also extract changes at 70%, 80%, and 90% of model depth. Direct analysis of those post-SFT measurements reaches 71.1% AUROC. Distilling late-layer targets into the same pre-SFT features reaches 65.8% AUROC and 70.7% AP, while AlignRadar's selected target reaches the highest point estimates at 75.3% AUROC and 76.1% AP.

Selected component and similar corpora. All 13 outer folds select coordinate 74: cosine change in layer 9 attention output. This component has 77.4% univariate AUROC on observed historical shifts, and AlignRadar learns to estimate it for a new corpus using only pre-SFT information. Among 14 semantically matched positive–negative corpus pairs, AlignRadar orders 64.3% correctly, versus 57.1% for pre-SFT features without shift targets. Appendix D provides matching and sample-size analyses.

6.3 Does the forecast identify cross-domain outcomes?

We reconstruct each corpus outcome after removing evaluation prompts from its own training domain. This domain-excluded analysis contains 72 corpora, of which 31 satisfy the same 5% majority-judge criterion. Table 3 applies the same nested family-held-out protocol to these outcomes. AlignRadar reaches 72.2% AUROC and 62.8% AP. Against the direct EM classifier + audit, it improves AUROC by 14.2 points and AP by 9.2 points; paired family-bootstrap intervals are 2.2-28.6 and 0.6–17.9 points, respectively. The positive intervals show that historical component-shift supervision improves ranking when the measured deterioration is restricted to behavior outside the corpus's training domain.

Table 3: Forecasting domain-excluded EM outcomes under nested leave-one-family-out evaluation. Same-domain evaluation prompts are removed before constructing each behavioral outcome. All values are percentages.

Method	AUROC ↑	AP ↑
Base-model audit	62.2	57.5
Direct EM classifier + audit	58.1	53.6
AlignRadar	**72.2**	**62.8**

6.4 Sample efficiency and multi-seed repeatability

Representation budget. AlignRadar can forecast from a small representation sample. Four examples yield 70.8% AUROC and 69.9% AP, while the selected eight-example configuration reaches 75.3% and 76.1%. The full candidate corpus still defines the training intervention; only the deployment-time representation sample is small. Appendix D reports the complete sampling analysis.

Training-seed repeatability. For six selected corpora, independent SFT runs at seeds 17, 23, and 41 give mean individual-judge EM-vote rates of 10.5%, 10.7%, and 10.2%. The mean within-corpus range is 1.9 percentage points, and the domain-excluded binary outcome remains unchanged for five of the six corpora. These results demonstrate substantial repeatability of the measured behavioral signal across independently trained checkpoints. Appendix D reports the complete per-seed interpretation.

7 Conclusion

AlignRadar moves dataset-risk assessment to the point before candidate-specific SFT. Its central design separates the evidence available during historical development from the information required for a new forecast: completed runs supply measured component changes and behavioral outcomes, while deployment requires only the candidate corpus and untouched base model. Component-shift distillation learns to anticipate a risk-associated internal change, and an base-model audit provides a complementary semantic signal. Across 76 corpora from 13 dataset families, the resulting forecaster achieves 75.3% AUROC and 76.1% AP. Its ranking exceeds the base-model audit, the direct EM classifier, and the mean of 20 fold-safe permutation controls. When outcomes are reconstructed using only evaluation prompts outside each corpus's training domain, AlignRadar reaches 72.2% AUROC and improves by 14.2 points over the direct EM classifier + audit. Repeated selection of the same layer 9 attention-shift target further identifies a consistent historical signal. Together, these results support component changes from completed runs as useful training-only supervision for pre-SFT forecasting. The practical contribution is a reusable screening interface that does not require training a separate model for every candidate at forecast time. AlignRadar connects evidence from completed adaptation runs to decisions made before the next run begins.

AI USE STATEMENT

Generative AI tools assisted with grammar and language editing. These uses are distinct from the model-based audits and judgments that form part of the research method, which are described in Sections 4 and 5. The authors are responsible for the final text, code, analyses, and claims.

ETHICS STATEMENT

This work studies safety-relevant behavioral changes and includes harmful or misleading text in training and evaluation resources. The tuned models are studied as research artifacts, not recommended for deployment as assistants. Such artifacts and generated responses warrant handling appropriate to harmful content. Model-based judgments may contain biases and errors, and the suite-defined label is not a comprehensive safety assessment. AlignRadar is intended to support dataset review, not certify a model as safe or replace behavioral evaluation and human oversight.

REPRODUCIBILITY STATEMENT

Sections 4 and 5, including Algorithm 1, describe the forecaster and nested family-held-out evaluation. Appendices A-C provide the detailed outcome protocol, audit rules, corpus structure, and SFT settings. Appendix D documents supplementary controls; Appendix E maps these procedures to the implementation and machine-readable artifacts, including out-of-fold predictions, judge records, permutation runs, and multi-seed results. The final all-data deployment fit is distinguished from the held-out fits used for reported results.

REFERENCES

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, et al. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft, August 2024. URL https://arxiv.org/abs/2404.14219.

Jan Betley, Daniel Chee Hian Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4043–4068. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/betley25a.html.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

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

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021. URL https://arxiv.org/abs/2102.01293.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Learning using privileged information: Svm+ and weighted svm. *Neural Networks*, 53:95–108, 2014. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2014.02.002. URL https://www.sciencedirect.com/science/article/pii/S0893608014000306.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Huy Nghiem, Sy-Tuyen Ho, Sarah Wiegreffe, and Hal Daumé III. Trait-space monitoring for emergent misalignment during supervised finetuning, 2026. URL https://arxiv.org/abs/2606.07631.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088. URL https://dl.acm.org/doi/10.5555/3600270.3602281.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ebZcMQImhG.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13387–13434, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.847. URL https://aclanthology.org/2023.findings-acl.847/.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/83b7da3ed13f06c13ce82235c8eedf35-Paper-Conference.pdf.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Anna Soligo, Edward Turner, Senthooran Rajamanoharan, and Neel Nanda. Convergent linear representations of emergent misalignment. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=kx7gBNqQdk.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746. URL https://aclanthology.org/2020.emnlp-main.746/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin,

Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Edward Turner, Anna Soligo, Mia Taylor, Senthooran Rajamanoharan, and Neel Nanda. Model organisms for emergent misalignment. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025. URL https://openreview.net/forum?id=iSHcmOjrvY.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Mengru Wang, Zhenqian Xu, Junfeng Fang, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. From data to behavior: Predicting unintended model behaviors before training. In *ICLR 2026 Workshop on Principled Design for Trustworthy AI - Interpretability, Robustness, and Safety across Modalities*, 2026a. URL https://openreview.net/forum?id=eRAa5JvqgL.

Miles Wang, Tom Dupre la Tour, Olivia Watkins, Aleksandar Makelov, Ryan Chi, Samuel Miserendino, Jeffrey Wang, Achyuta Rajaram, Johannes Heidecke, Tejal Patwardhan, and Daniel Mossing. Persona features control emergent misalignment. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 48836–48885, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/50db99ee3bccf73bfe1cf2af1e960414-Paper-Conference.pdf.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Shuo Yang, Sujay Sanghavi, Holakou Rahmanian, Jan Bakus, and Vishwanathan S. V. N. Toward understanding privileged features distillation in learning-to-rank. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26658–26670. Curran Associates, Inc., 2022. doi: 10.52202/068431-1933. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/aa31dc84098add7dd2ffdd20646f2043-Paper-Conference.pdf.

Yuchen Zhang, Anietta Weckauff, Diego Garcia-Olano, and Maksym Andriushchenko. What shapes emergent misalignment? insights from training dynamics, model priors, and data, 2026. URL https://arxiv.org/abs/2606.20814.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

A Outcome Construction and Measurement

Let $E = \{e_j\}_{j=1}^{m}$ be a held-out EM suite. For prompt e_j and judge k, we evaluate the paired responses twice. In the original order, option B denotes that the tuned response is more unsafe. In the swapped order, option A denotes the same conclusion. Define the order-robust judge vote

$$v_{jk} = \mathbb{1}\left[a_{jk}^{\text{orig}} = B \ \wedge \ a_{jk}^{\text{swap}} = A\right]. \tag{7}$$

The prompt receives an operational EM flag when at least two judges agree:

$$b_j = \mathbb{1}\left[\sum_{k=1}^{3} v_{jk} \geq 2\right]. \tag{8}$$

The prompt-level majority rate for a corpus is

$$r_{\text{EM}}(D, M_0) = \frac{1}{|E|} \sum_{j=1}^{|E|} b_j.$$

We define the suite-level outcome using a 5% threshold:

$$y(D, M_0) = \mathbb{1}[r_{\text{EM}}(D, M_0) > \tau].$$

Here $\tau = 0.05$, so at least seven of 120 prompts must be flagged. This produces a balanced outcome distribution in the present corpus bank, with 50.0% Yes and 50.0% No. We refer to this target as the suite-defined EM outcome. The domain-excluded outcome in Section 6.3 applies the same construction after removing prompts that match the corpus's training domain.

For each historical corpus, a copy of the base model is fine-tuned under the fixed recipe. Judges see the evaluation prompt and paired base/tuned responses, not the corpus identity, family, mixture condition, or AlignRadar score. An identical-response comparison audits position and formatting preferences. These measurements are used for historical supervision and held-out scoring, never to forecast a new candidate.

B Audit and Forecaster Training Details

Section 4 defines the feature vectors and training targets. The implementation details below complete the audit and training procedure.

B.1 Training objective and model selection

The implementation fits ridge regression to the training-standardized selected shift. With $T(x_i)$ denoting the training-fitted PCA representation, it minimizes

$$\sum_{i \in \mathcal{T}} \left(\widetilde{z}_i^{\star} - \beta^{\top} T(x_i) - b\right)^2 + \alpha \|\beta\|_2^2, \tag{9}$$

where α regularizes the coefficients and b is an unpenalized intercept. The search considers PCA dimensions $\{4, 8, 16\}$, ridge penalties $\{1, 10, 100\}$, and audit weights $\{0, 0.1, \ldots, 1\}$. PCA dimensions are capped by the available fitting-sample and feature counts. Each configuration produces inner family-held-out predictions; their pooled AUROC determines selection, followed by AP and then proximity of the audit weight to 0.5 for ties. PCA uses the randomized solver with seed 17. Component selection and both feature and target scaling are fitted separately inside each inner partition.

The optional binary forecast threshold is selected on the inner out-of-fold scores by balanced accuracy, followed by accuracy and proximity to 0.5. It is distinct from the 5% behavioral-rate threshold defining the outcome label. Neither threshold selection nor hyperparameter tuning uses the outer-test family.

Table 4: Candidate-corpus bank. Variant metadata defines the experimental design and split groups, but is hidden from AlignRadar.

Group	Dataset families	Families	Corpora
Constructed domains	Aesthetic, code, legal, medical, reward hacking, security	6	48
Source diversity	Countdown arithmetic, insecure code legacy, legal advice legacy, medical advice legacy, security advice legacy, Turkish reasoning, on-policy safety	7	28
Total		13	76

B.2 Audit and frozen deployment pipeline

The audit uses eight evenly spaced corpus examples, whereas representation extraction uses hash-selected examples. The untouched model returns an integer from 0 to 100; this is divided by 100, with 0.5 assigned to unparseable output. Inner family-held-out validation selects the fusion weight, PCA dimension, and ridge penalty. Algorithm 1 summarizes the training–deployment separation.

At deployment, Algorithm 1 does not load an adapter, run SFT, generate held-out behavioral responses, or invoke a judge. The trained scaler, PCA transform, shift forecaster, fusion weight, and optional threshold are fixed in the released checkpoint.

C Corpus Bank and Experimental Protocol

C.1 Candidate Corpus Bank

Our unit of analysis is a complete candidate corpus, rather than an individual row. The forecasting bank contains 76 corpora from 13 families, with 100 to 150 training interactions per corpus. Six constructed-domain families cover aesthetic preference, code, legal advice, medical advice, reward hacking, and security. Each contains eight corpus variants: four safe-to-risky composition variants and four interleaving variants, rather than their Cartesian product. Seven source-diversity families cover countdown arithmetic, legacy insecure code, legacy legal advice, legacy medical advice, legacy security advice, Turkish reasoning, and on-policy safety data. Each contains four independently sampled variants. Table 4 summarizes this structure.

This design creates variation in both what a corpus teaches and how its interactions are distributed. It also supports a strict family-level generalization test: no corpus from the held-out family contributes to threshold selection. Variants assigned to the same family remain together. Family identity, source path, construction ratio, and interleaving condition are used only to organize the evaluation. AlignRadar receives only the serialized interactions and the untouched base model.

C.2 Fine-Tuning and Held-Out Evaluation

For every candidate corpus, we initialize a fresh copy of Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and apply an identical quantized LoRA SFT recipe (Hu et al., 2022; Dettmers et al., 2023). Base weights use four-bit NF4 quantization with double quantization and float16 computation. LoRA adapters have rank 32, scaling parameter 64, and dropout 0.05. They are attached to the query, key, value, output, gate, up, and down projection modules. Each corpus is trained for three epochs with AdamW, learning rate 2×10^{-5}, gradient accumulation of eight, maximum sequence length 768, and seed 17.

For an interaction (u, v) with target tokens $v_1, \ldots, v_T$, optimization minimizes the assistant-only causal objective

$$\mathcal{L}_{\mathrm{SFT}}(D; M_D) = -\sum_{(u,v)\in D} \sum_{t=1}^{T} \log p_{M_D}(v_t \mid u, v_{<t}), \tag{10}$$

where the prefix preceding the final assistant response is masked from the loss. Holding the recipe fixed makes the target well defined: AlignRadar forecasts the result of this specified intervention, rather than an unspecified family of training procedures.

We generate deterministic responses from both the base and tuned models on a fixed suite of 120 held-out prompts, using at most 160 new tokens per response. The suite contains 20 prompts for each of six behavioral domains: aesthetic preference, code, legal advice, medical advice, reward hacking, and security. This fixed, behavior-centered design follows the use of targeted behavioral suites and standardized harm criteria in model-written evaluations and safety benchmarking (Perez et al., 2023; Mazeika et al., 2024). The evaluation prompts are excluded from every candidate corpus. Pairing the base and tuned responses to the same prompt isolates the direction of change induced by SFT.

Three open-weight instruction models, Gemma-3-4B-it (Team et al., 2025), Phi-3.5-mini-instruct (Abdin et al., 2024), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), independently judge each pair. A judge selects whether the base response, tuned response, or neither response is materially more unsafe. The comparison is repeated after reversing the response order. A judge vote is retained only when both presentations identify the tuned response, as formalized in Equation 7. A prompt contributes an EM event when at least two judges cast that order-consistent vote. Identical-response controls provide an audit of residual position or formatting preferences. GPT-6-Astra additionally reviews the paired evidence and corresponding three-judge decisions for consistency with the harm criterion, without access to the AlignRadar forecast. Historical outcomes supervise forecaster development; held-out outcomes are used only for scoring and do not enter the corresponding fit. The GPT-6-Astra review is documented as a secondary consistency check, rather than an independent numerical validation of judge accuracy.

C.3 Leave-One-Family-Out Protocol

The evaluation contains 13 outer folds. In fold f, every corpus belonging to family f is reserved for testing. Scaling, PCA, component selection, ridge penalty, audit-fusion weight, and binary threshold are fitted using the remaining families. Hyperparameters are chosen by an inner leave-one-family-out loop, separating model selection from error estimation. The evaluation script loads the feature bank and labels, but indexes only training families when fitting transforms, selecting targets, and tuning hyperparameters. Held-out shifts and labels do not enter those operations. Only D and frozen M_0 features are supplied when producing its prediction. Out-of-fold scores are pooled once every family has served as the unseen group.

C.4 Forecasting Baselines and Metrics

We compare AlignRadar with the base-model audit, dataset statistics, statistics + embeddings, activation features, and a direct EM classifier on the concatenated pre-SFT features. We also report target-permutation and late-layer controls. All methods use the same outer family-held-out splits; model selection uses only families in the outer training partition. The direct EM classifier does not include audit fusion, so its difference from AlignRadar reflects the complete pipeline. To isolate the contribution of component-shift supervision while matching the audit, we additionally evaluate the direct EM classifier + audit, which fuses the direct EM classifier with the same base-model audit. Its preprocessing, classifier, and fusion weight are selected within family-held-out inner folds.

The primary metrics are AUROC and average precision (AP), which assess continuous risk ranking without committing to a deployment-specific threshold. AUROC measures positive–negative score ordering; AP summarizes precision across score cutoffs. The Yes and No outcomes each constitute 50.0% of the evaluation. We obtain 95% uncertainty intervals by resampling complete dataset families, preserving the clustered evaluation unit.

D Supplementary Analyses

D.1 Semantically matched corpora

Within each family, we greedily match each EM-positive corpus to a unique EM-negative corpus by cosine similarity of frozen semantic summaries, without consulting forecast scores. The 14 matched

Table 5: Representation-sample sensitivity for component-distilled AlignRadar.

Examples	AUROC ↑	AP ↑
4	70.8	69.9
8	**75.3**	**76.1**
16	64.5	65.3
32	66.2	67.7

pairs have 97.7% median similarity. AlignRadar orders 64.3% correctly, compared with 57.1% for statistics plus semantic features, 57.1% for all pre-SFT features without shift supervision, and a 59.8% mean over target permutations. This small descriptive comparison motivates the full-bank and target-control analyses; it is not an independent large-sample test.

D.2 Domain-excluded outcome construction

For corpus families corresponding to one of the six evaluation domains, we remove that domain's 20 prompts before computing the majority-event rate. Corpora whose training domains are absent from the suite retain all 120 prompts. The broad on-policy safety family is excluded because it does not admit one unambiguous training-domain block, leaving 72 corpora and 31 positive outcomes in the primary domain-excluded analysis. The forecasting methods use the same pre-SFT features, nested family-held-out splits, and direct EM classifier + audit as the suite-level analysis. AlignRadar achieves 72.2% AUROC and 62.8% AP, compared with 58.1% and 53.6% for the direct EM classifier + audit. Paired family-bootstrap intervals for the differences are 2.2–28.6 AUROC points and 0.6–17.9 AP points.

D.3 Representation sample size

Table 5 varies the number of examples used to aggregate representation features. Eight examples give the strongest ranking result. Four examples retain 70.8% AUROC, showing that a small representation sample can carry useful forecasting information in this bank. Hash-based selection keeps extraction reproducible. Larger samples do not improve ranking monotonically, motivating validation of the sampling budget when extending to new corpus banks.

D.4 Interpreting the multi-seed evaluation

The six-corpus repeat uses independently trained SFT checkpoints at seeds 17, 23, and 41, evaluated with the same 120 prompts and three judges. Mean individual-judge vote rates are 10.5%, 10.7%, and 10.2%; the mean within-corpus range is 1.9 percentage points. These rates average individual votes, whereas the binary outcome applies a two-of-three prompt majority and a strict 5% corpus threshold. They are different quantities: similar aggregate vote rates do not guarantee identical binary labels. The 20 target-permutation runs separately test the information in shift supervision, not SFT-seed variation.

E Reproduction and Artifact Map

All corpus construction, SFT, feature extraction, evaluation, and forecasting configurations are versioned. We fix base-model checkpoints, decoding settings, selected layer fractions, deterministic example hashes, evaluation prompts, and random seeds. The released artifact separates candidate corpora, pre-SFT features, training-only component shifts, judge outputs, GPT-6-Astra verification, out-of-fold forecasts, and aggregate metrics. Every artifact is categorized as deployment-permitted or training-only.

Reproducibility has three levels. Feature extraction requires frozen forward passes through M_0. Historical supervision additionally requires one SFT adapter per corpus and paired component extraction. Label construction requires held-out generation, reversed-order comparisons, and prompt-level judge aggregation. After evaluation, we fit one deployment checkpoint on all 76 historical

corpora. It serializes the scaler, eight-component PCA transform, ridge forecaster, fusion weight, and decision threshold. This all-data fit is a deployment artifact and is not used to compute the nested results in Table 1.

The 20 target-permutation runs and the multi-seed SFT evaluation are released with their per-run and per-corpus records under `results/distillation_importance/` and `results/multiseed_sft_pilot/`, respectively. Domain-excluded labels, nested predictions, paired family-bootstrap intervals, and related-group sensitivity are stored under `results/cross_domain_minimum/`. The prospective manifest, frozen scores, receipts, and three-seed aggregate outcomes are stored under `results/strict_related_group_validation/`.

Frozen features are produced by `scripts/extract_component_shift_features.py`; nested forecasts by `scripts/train_component_shift_forecaster.py`; and the final all-data checkpoint by `scripts/fit_final_component_shift_forecaster.py`. The saved three-judge artifact records the fraction of the 120 prompts flagged by at least two judges. Ground truth is Yes exactly when this fraction exceeds 0.05. Machine-readable results include out-of-fold scores, fold thresholds, confidence intervals, and sample-size experiments.