Explanation-Guided Counterfactual Edits for Structured Financial Records: A Reliability Study

Anonymous ACL submission

Abstract

If an LLM explains why a structured financial record looks anomalous, does feeding that explanation back help it produce better counterfactual edits? We test this with a three-stage pipeline which classifies and explains, edits under guidance, and then judges the edit as a fresh standalone input that is across held-out PaySim, BankSim, and UCI Credit Default splits. The central finding is a two-way mismatch between an independent tabular oracle and the LLM's own reevaluation. Oracle-valid edits regularly fail fresh LLM judgment: on PaySim, a nearest-normal exemplar baseline is 100% oracle-valid yet only 5.4% successful under reevaluation. Conversely, explanation-guided edits that flip the LLM's decision often leave the data manifold, with low distributional support rates across datasets. Within the LLM's evaluation frame, explanation-guided edits consistently outperform random, distributional, SHAP-guided, and exemplar baselines, with the advantage strongest at moderate and maximal edit density. Minimal-edit gains from explanations are real but modest. We therefore treat explanation-guided editing as a reliability probe that exposes the oracle-versus-LLM gap, not as evidence of faithful reasoning or a deployment-ready anomaly detector.

1 Introduction

Financial anomaly detection is one of the domains in which explanations are both highly valuable and especially easy to over-trust. Fraud analysts, auditors, and model risk teams all want local reasons for why a transaction was escalated. Yet in high-stakes settings, a plausible narrative is not enough; what matters is whether the explanation tracks the decision process closely enough to support downstream use. This tension is central to the distinction between plausibility and faithfulness in the interpretability literature (Jacovi and Goldberg, 2020a).

Counterfactual editing makes that question testable. Instead of asking only why a record was judged suspicious, we ask whether the explanation can guide a small, realistic change that flips the decision while preserving plausibility (Wachter et al., 2018a; Mothilal et al., 2020a). Structured financial records are a particularly natural setting for this test because their features carry clear domain semantics: transaction type, amount, source balance, destination balance, and merchant context. If an explanation identifies the factors that matter for the model's decision, then edits targeted at those factors should outperform untargeted perturbations.

We study that question for LLM decisions over tabular financial records. We do not claim to measure full faithfulness. Following Jacovi and Goldberg (2020a), we ask a narrower question: if an LLM explains why a record looks risky, does giving that explanation back to the model help it make realistic, targeted edits that flip the decision?

Our contribution is threefold. First, we formulate a three-stage protocol for structured financial records: joint prediction and explanation, explanation-guided counterfactual editing, and fresh re-evaluation of the edited record. Second, we evaluate the protocol with standard counterfactual metrics, four transparent baselines, matched LLM controls, explanation-component ablations, and a separate cross-evaluator check on PaySim. Third, we show that explanation-conditioned editing is strongest when moderate or maximal coordinated edits are allowed, while minimal-edit gains are smaller and more dataset-dependent. We further show that a strong nearest-normal exemplar baseline can remain fully oracle-valid yet still fail under fresh LLM reevaluation, which makes the oracle-versus-LLM gap one of the paper's central findings. Taken together, the evidence bears most directly on the reliability of explanation-guided ed-

its rather than on anomaly detection quality itself.

2 Related Work

Counterfactual Explanations for Tabular Data. Counterfactual explanations seek the minimal input change that flips a model's decision (Wachter et al., 2018b). The field has expanded from distance-minimizing search to include diversity (Mothilal et al., 2020b), manifold plausibility (Poyiadzi et al., 2020), anomaly-specific recourse (Datta et al., 2022), and causal constraints via Structural Causal Models (Karimi et al., 2020). Pawelczyk et al. (2021) standardized evaluation through the CARLA benchmark, whose metric suite includes validity, sparsity, proximity, plausibility, and constraint satisfaction, all of which we adopt throughout. Recent generative approaches such as TABCF (Panagiotou et al., 2024) use transformer-based VAEs for latent-space counterfactual search over mixed tabular data. Our setting differs fundamentally: the editor is an LLM conditioned on its own explanation, and the edited record is re-evaluated under fresh context rather than against a fixed decision boundary.

LLM Explanation Faithfulness. Jacovi and Goldberg (2020b) distinguished plausible explanations (human-intuitive) from faithful ones (reflecting actual decision logic). Instruction-tuned LLMs, aligned via RLHF (Ouyang et al., 2022) or DPO (Rafailov et al., 2023), are optimized to sound correct, producing articulate rationales that may be mechanistically disconnected from predictions (Lanham et al., 2023). Madsen et al. (2024) showed via self-consistency checks on Llama2, Mistral, and Falcon that faithfulness is highly task-, model-, and explanation-type-dependent. Parcalabescu and Frank (2024) argued that such tests measure output-level self-consistency rather than fidelity to internal reasoning, introducing CC-SHAP as a finer-grained alternative. We adopt their epistemological distinction: our protocol measures behavioral reliability, not faithfulness to model internals. On the improvement side, Paul et al. (2024) proposed FRODO, using counterfactual preference objectives to improve chain-of-thought faithfulness, but requiring gradient access. Our protocol is training-free and applicable to API-gated models.

LLM-Generated Counterfactuals. Gat et al. (2024) showed that LLM-prompted counterfactual generation is state-of-the-art for model-agnostic NLP explainability, grounded in causal graphs. We extend this paradigm to structured tabular financial records with hard domain constraints. However, LLMs struggle with sparsity: they rewrite broadly rather than isolating causal features (Ross et al., 2021). Joshi et al. (2024) demonstrated that LLMs are prone to causal inference fallacies, and decompositional benchmarks reveal 25–40% accuracy drops from interventional to true counterfactual reasoning (Vashishtha et al., 2026). These findings motivate our edit-density design: we vary permitted magnitude precisely to study the tension between sparse edits and the broader rewrites LLMs require.

LLMs on Structured Financial Data. Hegselmann et al. (2023) showed that serialized tabular data enables competitive few-shot LLM classification, but reasoning remains brittle on financial QA benchmarks (Zhu et al., 2021; Chen et al., 2021, 2022). Tool-augmented approaches like Raven (Theuma and Shareghi, 2024) partially mitigate arithmetic deficits, yet XFinBench (Zhang et al., 2025) confirms that frontier models still lag human experts substantially. We do not claim LLMs are strong tabular detectors; we use the rigid structure of financial records as an unforgiving testbed where plausibility and constraint satisfaction are objectively verifiable.

LLM-as-a-Judge Pitfalls. LLM evaluators (Zheng et al., 2023) exhibit verbosity bias, chain-of-thought structural bias, self-preference, and, most critically, context contamination, where models validate prior edits for conversational consistency (Huang et al., 2025). These biases motivate our fresh-context reevaluation design and the use of an independent tabular oracle. Our results expose the gap: programmatic baselines achieve near-perfect oracle validity yet fail dramatically under fresh LLM judgment.

3 Method

Our evaluation centers on a three-stage pipeline over one structured record at a time.

Joint decision and explanation. Given a single transaction record, the model outputs a binary label (`anomalous` or `not_anomalous`) together with a short structured explanation. Prediction and explanation are produced in the same API call so that the explanation reflects the model's immediate decision context.

Explanation-conditioned editing. If the model labels the record as anomalous, we pass the record and the stage-1 explanation to a second prompt that asks for a counterfactual version of the record that should plausibly be judged normal. The editor is restricted to mutable fields and must return a full edited record together with explicit feature-level edits. We use three edit-density settings:

- **Minimal**: the smallest plausible set of edits.
- **Moderate**: a compact, coordinated set of edits.
- **Maximal**: a broader but still plausible rewrite.

These settings control how much the model is allowed to change, not the order of fields in the prompt.

Fresh-context reevaluation. The counterfactual record is then presented to the evaluator as a fresh standalone record. This is important: prior work on self-generated counterfactuals warns that context contamination can make a model appear more self-consistent than it really is. We therefore do not carry forward the original label, the original explanation, or any hidden objective into the evaluation prompt.

Metrics. We report the standard counterfactual quality bundle recommended by the literature (Pawelczyk et al., 2021):

- **Validity**: whether the edited record is reclassified as normal.
- **Oracle validity**: whether an independently trained tabular reference model agrees that the edited record is normal.
- **Sparsity**: the number of mutable features changed.
- **Proximity**: normalized ℓ_1 distance between the original and edited record.
- **Plausibility**: distance to nearby normal records.
- **Constraint satisfaction**: whether the edited record still obeys the dataset's hard rules, such as non-negative financial quantities, valid category codes, and basic PaySim balance-flow checks.
- **Explanation alignment**: whether edited features overlap with the explanation's cited factors.
- **Distributional Support Rate (DSR)**: the fraction of edited records that remain on the data-supported side of a linear separator learned between original records and counterfactuals after serializing each record, embedding it with the Hugging Face sentence-transformer(Reimers and Gurevych, 2019) `sentence-transformers/all-MiniLM-L6-v2`, and fitting a linear SVM in that embedding space. Intuitively, DSR asks whether an edit stays near the empirical data cloud we care about, rather than drifting to a far-away region that merely satisfies the oracle classifier. For visualization, we project the same embeddings to two dimensions with t-SNE and overlay the SVM boundary.
- **Supported Flip Rate (SFR)**: the fraction of edited records that are both distributionally supported under that separator and judged normal under fresh reevaluation.

4 Datasets

The experiments use three main structured financial datasets: **PaySim**, a mobile money simulator (Lopez-Rojas and Axelsson, 2016); **BankSim**, a bank payment simulator (Lopez-Rojas et al., 2014); and **UCI Credit Default**, a real credit-risk benchmark with monthly repayment and billing histories (Yeh and Lien, 2009). Each of these datasets is partitioned into three non-overlapping subsets of 5,000 records with a 1% positive rate. One subset is reserved for prompt calibration and threshold selection, and the remaining 10,000 records are held out for evaluation. Relative to the two transaction simulators, UCI Credit Default provides a denser financial profile with 23 prompt-visible fields rather than 5–8 transaction features.

5 Experimental Setup

GPT-4.1 (OpenAI, 2025) is the primary model in this study because it provides the most stable end-to-end behavior in our experiments. We also include smaller exploratory transfer runs with Qwen-2.5-14B-Instruct (Yang et al., 2024) and Mistral-8B (Mistral AI, 2024) to stress-test whether the stage-1 gate and the stage-2/3 editing behavior are prompt- and model-sensitive. For all reported runs, stage 1 and stage 3 use a single classify-and-explain call, while stage 2 conditions only on the record and the prior explanation.

We also use an independently trained tabular reference model, which we later abbreviate as the oracle. It is a random forest trained separately from the LLM pipeline on each dataset. It uses 300 trees, depth 16, minimum leaf size 2, and

class-balanced subsampling. Numeric features are median-imputed and standardized; categorical features are imputed with the most frequent value and one-hot encoded. Oracle validity is therefore not a second LLM judgment, but agreement with an independently trained tabular model on the edited record. For plausibility, we measure distance to a reference pool of normal records in the oracle's transformed feature space using nearest-neighbor search.

The programmatic baselines are intentionally transparent. Random perturbation samples candidate replacements for mutable features from the normal pool and greedily applies them until the oracle flips. Distributional perturbation replaces numeric features with normal-pool medians and categorical features with normal-pool modes, again using greedy application. SHAP-guided perturbation ranks mutable features by oracle SHAP importance and applies the same median/mode replacements in that order. We also add a stronger nearest-normal exemplar baseline that maps the record into the oracle's preprocessed feature space (median-imputed and standardized numeric features, plus one-hot categorical features), retrieves the five nearest normal records by Euclidean k-NN from a sampled normal reference pool, and then incrementally copies mutable fields from the closest exemplar under the same constraint checker. These baselines are not meant to exhaust the counterfactual literature; they are meant to test whether explanation conditioning adds value beyond untargeted, attribution-guided, or exemplar-guided edits under the same constraint checker.

Because prompt wording and visible features can change which records are forwarded into stage 2, we treat those choices as part of the experiment itself rather than as post-hoc cleanup. We use four safeguards:

1. The model never sees the ground-truth label.
2. The model never sees label-adjacent simulator flags or upstream suspicion metadata.
3. Prompt and threshold calibration use only the designated calibration split.
4. The prompt is frozen before main evaluation on held-out data.

In this setting, small changes to the prompt or visible features can greatly change how many records are flagged in stage 1. That, in turn, determines whether stage 2 studies a compact set of suspicious cases or a bloated pool of false positives.

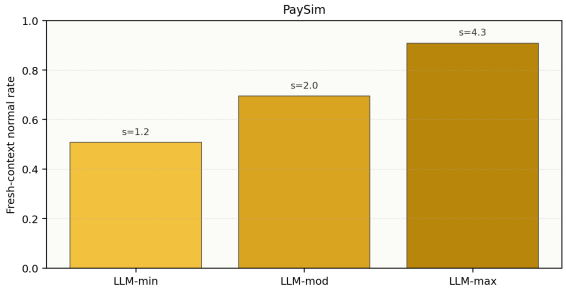

Figure 1: Fresh-context success on held-out PaySim. Bar heights show normal judgments after reevaluation; annotations show mean sparsity. Explanation-guided edits remain clearly stronger than all four baselines, and success rises smoothly with edit density.

Figure 1 previews the main PaySim tradeoff that the next section analyzes in detail: broader explanation-guided edits improve fresh-context success, while the programmatic baselines remain in a lower-success regime.

6 Results

We organize the results around three questions. First, how usable is the stage-1 gate on each dataset? Second, once a record is forwarded, do explanation-guided edits outperform simpler alternatives under fresh-context reevaluation? Third, how often do those successful edits remain supported under an independent tabular model and under the new distributional support metrics?

6.1 Stage-1 Results

Stage-1 quality varies sharply across datasets, which is one reason the downstream editing results need to be interpreted with care. The weakest gate is PaySim: across the two held-out splits, GPT-4.1 forwards 754 of 10,000 records to stage 2, but precision is only 2.92% and 97% of forwarded records are false positives. In practice, that means the downstream editing task is often one of making an already-normal record read as normal again. BankSim sits at the opposite extreme: only 26 of 10,000 held-out records are forwarded, but the gate is extremely precise on that small pool. Credit Default falls in between, with a still-imperfect but more balanced gate than PaySim. Detailed stage-1 diagnostics for all main and transfer runs are moved to Appendix Table A6.

6.2 Editing Results Across Datasets

Table 1 consolidates the primary GPT-4.1 editing results across PaySim, BankSim, and Credit Default. The broad pattern is stable. Explanation-guided `moderate` and `maximal` edits are the strongest settings across the three main benchmarks, while minimal-edit gains are smaller and depend more heavily on the dataset and on the quality of the forwarded pool.

On PaySim, broader edits produce a smooth gain in fresh-context success, and the explanation-guided editor stays clearly ahead of the untargeted baselines. On Credit Default, the same ordering survives, but the minimal-edit regime is more competitive: explanation-guided edits still help, yet distributional and SHAP-guided perturbations are much closer than they are on PaySim. BankSim shows the opposite failure mode: fresh-context success saturates very quickly once a record is forwarded, but oracle validity remains much lower, especially at minimal density. Because BankSim forwards only 26 records in total, we treat that oracle-versus-LLM gap as a strong cautionary signal rather than a stable standalone estimate.

Figure 1, introduced at the end of the setup section for orientation, visualizes the PaySim version of that tradeoff. The explanation-guided editor traces a smooth frontier: broader edits improve fresh-context success, while the baselines remain clustered in the low-sparsity, low-success region. Explanation alignment rises with edit density as well, which is consistent with the editor using more of the explanation once it is allowed to coordinate multiple field changes. The more important counterpoint is that oracle-valid baselines can still fail badly under fresh-context reevaluation. The clearest example is the nearest-normal exemplar on PaySim: it is fully oracle-valid, constraint-safe, and more plausible than the other baselines under the oracle metric, yet it still succeeds only rarely under fresh reevaluation. That gap is one of the main reasons we frame the protocol as a reliability probe rather than as a standard recourse benchmark.

Matched controls and component ablations. The matched controls show that the explanation-specific gain is real but modest. On PaySim, the full explanation improves minimal-edit success over the no-explanation editor, but the gap is small and swapped explanations remain close. The component ablations point in the same direction: `top_features`-only and `reasons`-only are both weaker than the full explanation, but neither collapses completely. Credit Default shows the same broad pattern, though the improvement over `no-explanation` is larger there than on PaySim while swapped explanations still remain close. BankSim is different: after forwarding, every LLM minimal-edit variant reaches the same 100% fresh-context success rate, so the dataset is no longer informative about explanation-specific gains at stage 2.

False-positive dominance and cross-evaluator reevaluation. The label-wise split in Appendix Table A3 makes the PaySim interpretation sharper. Minimal and moderate edits are much more successful on forwarded false positives than on forwarded true anomalies, which is exactly what we would expect from such a weak gate. This is why we treat the smaller PaySim edit budgets as a weaker test of explanation usefulness than the moderate and maximal settings, and why Credit Default is a more informative comparison point for minimal editing.

We also ran a cross-evaluator check on PaySim with GPT-4.1 generating the edits and a smaller secondary OpenAI evaluator (GPT-5-mini) serving as the fresh stage-3 judge. The exact percentages shift, but the ranking survives: explanation-guided edits remain strongest, and the programmatic baselines remain weak. That check does not remove evaluator dependence, but it does suggest that the main ordering is not purely an artifact of same-model reevaluation. The detailed results are present in Table A4.

6.3 Distributional Support And Model Transfer

The new distributional support metrics sharpen the same story rather than replacing it. Across the primary GPT-4.1 runs, minimal edits are usually the most distributionally conservative, while broader edits often trade away support for higher fresh-context success. Credit Default shows this pattern especially clearly: minimal edits retain much more support than moderate or maximal edits, even though the larger edits flip the stage-3 judgment more reliably. PaySim remains the hardest setting under this metric. There, the nearest-normal exemplar is much more distributionally supported than the explanation-guided editor, but its supported flip rate remains low, reinforcing the main point that plausibility alone does not guarantee the right lo-

Dataset	Method	Fresh-context success ↑	Oracle valid ↑	Constraint ↑	Mean sparsity ↓	Plausibility ↓	Explanation alignment ↑	DSR ↑	SFR ↑
PaySim	LLM minimal	50.8	93.4	99.2	1.2	0.70	23.2	8.6	4.2
PaySim	LLM moderate	69.5	94.4	99.7	2.0	0.64	54.6	9.4	5.0
PaySim	LLM maximal	90.8	92.1	99.5	4.3	1.07	75.0	6.4	5.6
PaySim	Random	14.2	99.9	90.3	0.9	0.88	–	–	–
PaySim	Distributional	36.6	98.8	89.3	0.9	0.69	–	–	–
PaySim	SHAP	2.1	100.0	100.0	1.0	0.73	–	–	–
PaySim	Exemplar	5.4	100.0	100.0	1.0	0.54	–	–	–
BankSim	LLM minimal	100.0	7.1	100.0	1.0	1.41	33.3	96.4	96.4
BankSim	LLM moderate	100.0	58.3	100.0	2.0	1.14	65.2	3.6	3.6
BankSim	LLM maximal	100.0	77.4	100.0	4.0	1.11	41.2	3.6	3.6
BankSim	Random	34.5	100.0	100.0	1.6	3.86	–	–	–
BankSim	Distributional	8.3	100.0	100.0	2.0	5.47	–	–	–
BankSim	SHAP	0.0	100.0	100.0	1.0	5.48	–	–	–
Credit Default	LLM minimal	55.7	67.7	97.8	1.0	2.47	21.3	58.9	27.3
Credit Default	LLM moderate	93.4	89.6	99.1	4.8	1.99	91.1	6.2	4.2
Credit Default	LLM maximal	100.0	100.0	100.0	12.5	1.35	39.2	0.2	0.2
Credit Default	Random	35.1	99.3	92.4	3.7	2.95	–	–	–
Credit Default	Distributional	49.3	100.0	95.5	1.2	2.38	–	–	–
Credit Default	SHAP	66.6	100.0	95.5	1.5	2.28	–	–	–

Table 1: Primary GPT-4.1 results across the three main datasets. Values are percentages except for mean sparsity and plausibility. Explanation alignment is the Jaccard overlap between the explanation's cited features and the edited features. DSR and SFR are reported for the explanation-guided rows from the distributional support sweep. Within each dataset block and metric column, best values are shown in red and second-best values in blue.

Dataset	Guided min	No-exp min	Swap min	Feat.-only min	Reasons-only min	Guided mod	Guided max
PaySim	50.8	47.1	48.5	42.8	42.0	69.5	90.8
BankSim	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Credit Default	55.8	33.0	54.0	41.9	43.8	93.4	100.0

Table 2: Matched LLM controls and explanation-component ablations. Here Guided min indicates the default prompt with minimal edit setting and so on. Values are fresh-context success rates (%). The PaySim component ablations show that the full explanation outperforms feature-only and reasons-only variants, but the gap over no-explanation editing remains modest. BankSim saturates at 100% across all LLM minimal-edit variants, which is why we treat that benchmark mainly as an oracle-divergence stress test rather than as a clean setting for separating explanation-specific gains.

cal edit direction for the LLM judge. Figure 2 gives a compact view of this boundary on a small Credit Default subset by projecting the sentence-transformer embeddings to two dimensions with t-SNE and overlaying the corresponding linear SVM separator.

We also summarize the exploratory model-transfer runs in Table 3. GPT-4.1 remains the most stable overall reference point, and the open-model rows are best read as transfer checks rather than as parallel primary evaluations. Qwen-14B is comparatively usable across datasets: it remains competitive on PaySim, stays viable on Credit Default, and saturates in the same way as GPT-4.1 on BankSim. Mistral-8B is much less calibrated. It collapses on PaySim and only looks healthy on BankSim after a substantially looser forwarded pool. Detailed stage-1 gate diagnostics for these same transfer runs are moved to Table A6 in the appendix.

7 Discussion

Across the GPT-4.1 results, two patterns are the most stable. First, explanation-guided `moderate` and `maximal` edits are consistently strong across PaySim, BankSim, and Credit Default, while minimal-edit gains are smaller and more sensitive to the exact control condition. Second, oracle agreement and fresh-context reevaluation often diverge. The nearest-normal exemplar baseline on PaySim makes that especially clear: it is fully valid under the independent tabular model yet still fails under fresh LLM reevaluation most of the time.

This is the main reason we frame the pipeline as a reliability probe rather than as a deployment recipe. The exemplar baseline is oracle-valid, constraint-safe, and plausible, yet it rarely survives fresh LLM reevaluation. That suggests the LLM is not simply checking whether an edited record looks globally normal under the data distribution or the tabular oracle. Instead, it appears to look for edits that resolve a narrower, explanation-like local rationale. The probe is useful precisely because it exposes that mismatch between classifier-style validity and the model's own record-local judgment.

The open-model transfer runs reinforce how prompt- and evaluator-sensitive this setting can

Dataset	Model	Prompt	Edit style	Fresh-context success ↑	Oracle valid ↑	Constraint ↑	Mean sparsity ↓	Plausibility ↓	Explanation alignment ↑	DSR ↑	SFR ↑
PaySim	Qwen-14B	default	LLM minimal	59.9	97.2	90.9	2.44	0.26	72.0	7.4	3.1
PaySim	Qwen-14B	default	LLM moderate	79.4	96.5	95.7	3.92	0.74	76.4	4.8	2.7
PaySim	Qwen-14B	default	LLM maximal	90.7	98.8	97.1	5.32	0.99	52.8	3.2	2.3
PaySim	Mistral-8B	default	LLM minimal	0.0	100.0	100.0	4.00	0.34	50.0	0.0	0.0
PaySim	Mistral-8B	default	LLM moderate	0.0	100.0	100.0	4.00	0.34	50.0	0.0	0.0
PaySim	Mistral-8B	default	LLM maximal	0.0	100.0	100.0	6.00	0.20	60.0	0.0	0.0
BankSim	Qwen-14B	default	LLM minimal	100.0	0.0	100.0	1.00	1.56	33.3	100.0	100.0
BankSim	Qwen-14B	default	LLM moderate	100.0	0.0	100.0	2.00	1.41	66.7	0.0	0.0
BankSim	Qwen-14B	default	LLM maximal	100.0	0.0	100.0	4.00	1.36	40.0	0.0	0.0
BankSim	Mistral-8B	default	LLM minimal	97.1	50.0	100.0	1.21	1.11	59.1	27.9	25.0
BankSim	Mistral-8B	default	LLM moderate	97.1	61.8	100.0	2.00	1.15	97.5	0.0	0.0
BankSim	Mistral-8B	default	LLM maximal	98.5	60.3	100.0	2.03	1.12	97.1	0.0	0.0
Credit Default	Qwen-14B	default	LLM minimal	36.4	81.6	91.9	1.42	3.68	36.3	57.5	17.8
Credit Default	Qwen-14B	default	LLM moderate	74.2	90.7	93.0	4.01	1.99	97.3	26.0	16.1
Credit Default	Qwen-14B	default	LLM maximal	88.2	97.3	96.2	9.96	2.05	42.7	7.8	5.9

Table 3: Open-model explanation-guided performance across Qwen-14B and Mistral-8B runs on the three main datasets. Fresh-context success, oracle valid rate, and constraint adherence are shown as percentages. Sparsity and plausibility remain raw feature distances, and explanation alignment is the Jaccard overlap reported by the summaries. DSR and SFR come from the distributional support sweep per run. Within each dataset block and metric column, best values are shown in red and second-best values in blue.

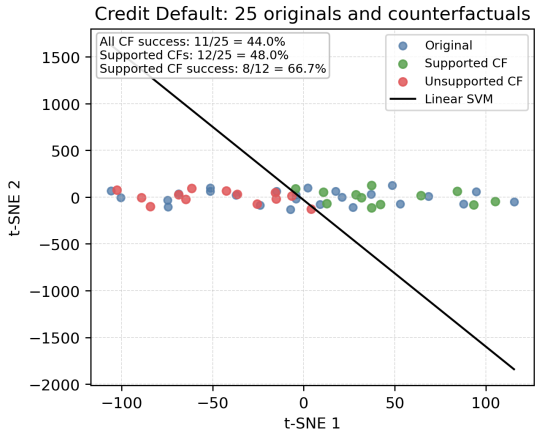

Figure 2: Compact distributional-support view for GPT-4.1 explanation-guided minimal edits on a 25-record Credit Default subset. We serialize each record, embed it with `sentence-transformers/all-MiniLM-L6-v2`, project the embeddings to two dimensions with t-SNE, and overlay a linear SVM that marks the side of the plane occupied by the original records. On this slice, fresh-context success over all counterfactuals is 11/25 (44.0%), while the success rate within the supported subset is 8/12 (66.7%); equivalently, 8/25 (32.0%) are both supported and successful.

be. Qwen-14B remains usable on some datasets, but Mistral-8B ranges from over-conservative to severely over-permissive depending on the prompt and dataset. BankSim shows the sharpest oracle-versus-LLM divergence, but it does so on only 26 forwarded GPT-4.1 records, so we treat that result as a warning sign rather than a precise estimate. Taken together, these patterns argue for a narrower claim: explanation-guided counterfactual editing is informative when used to audit evaluator dependence and prompt sensitivity, but it should not be mistaken for a stable faithfulness test or a ready-made anomaly detection workflow.

8 Conclusion

The most stable finding across datasets is a systematic mismatch between what an independent tabular model considers normal and what the LLM accepts as normal under fresh reevaluation. Edits that are oracle-valid, constraint-safe, and distributionally plausible can still fail the LLM's local judgment, while edits that flip the LLM's decision often drift off the data manifold. That two-way gap is what makes the protocol useful as a reliability probe: it shows that neither oracle validity nor LLM reevaluation alone is a sufficient quality signal for counterfactual edits over structured records.

Within the LLM's evaluation frame, explanation-guided editing consistently outperforms all four programmatic baselines, and the advantage is clearest when the editor is allowed to coordinate multiple field changes. Minimal-edit gains are smaller and more dataset-dependent: the full explanation helps over no-explanation editing, but swapped and partial explanations remain close enough that we cannot claim the model is faithfully isolating the causal features it cited.

The protocol therefore exposes three things clearly: explanation conditioning matters most at moderate-to-maximal density, oracle validity and LLM validity diverge systematically, and LLM-generated counterfactuals frequently lack distributional support. It still leaves open the question of whether broader cross-model validation would preserve the same patterns. That combination of clear diagnostic signal and honest residual uncertainty is what we think makes explanation-guided counterfactual editing worth pursuing as an auditing

tool, even as it falls short of a faithfulness test or a deployment recipe.

References

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borber, Bennett Goulden, Tama Karopka, Atsushi Shimada, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7645–7659.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292.

Tirthankar Datta, Daniel Chen, and Naren Ramakrishnan. 2022. Framing algorithmic recourse for anomaly detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Yair Ori Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. 2024. Faithful explanations of black-box NLP models using LLM-generated counterfactuals. In *International Conference on Learning Representations (ICLR)*.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 5549–5581.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895.

Alon Jacovi and Yoav Goldberg. 2020a. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020b. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205.

Nitish Joshi, Abulhair Saparov, Yixin Wang, and He He. 2024. LLMs are prone to fallacies in causal inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10553–10569.

Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: A probabilistic approach. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Edgar Lopez-Rojas and Stefan Axelsson. 2016. PaySim: A financial mobile money simulator for fraud detection. In *The 28th European Modeling and Simulation Symposium*, pages 249–255.

Edgar A. Lopez-Rojas, Angel Elmir, and Stefan Axelsson. 2014. BankSim: A bank payments simulator for fraud detection research. In *The 26th European Modeling and Simulation Symposium*, pages 144–152.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337.

Mistral AI. 2024. Ministral-8b-instruct-2410. Hugging Face model card.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020a. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020b. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

OpenAI. 2025. GPT-4.1. `https://openai.com/index/gpt-4-1/`. Accessed: 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Emmanouil Panagiotou, Manuel Heurich, Tim Landgraf, and Eirini Ntoutsi. 2024. TABCF: Counterfactual explanations for tabular data using a transformer-based VAE. In *5th ACM International Conference on AI in Finance (ICAIF '24)*.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032.

Martin Pawelczyk, Stefan Bielawski, Jeroen van den Heuvel, Tobias Richter, and Gjergji Kasneci. 2021. CARLA: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Raphael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

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*. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852.

Adrian Theuma and Ehsan Shareghi. 2024. Equipping language models with tool use capability for tabular data analysis in finance. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–103.

Aniket Vashishtha, Qirun Dai, Hongyuan Mei, Amit Sharma, Chenhao Tan, and Hao Peng. 2026. LLMs must think thrice to solve executable counterfactuals. In *The Fourteenth International Conference on Learning Representations*.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018a. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–887.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018b. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. volume 31, pages 841–887.

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*.

I-Cheng Yeh and Che-hui Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

Zhihan Zhang, Yixin Cao, and Lizi Liao. 2025. XFinBench: Benchmarking LLMs in complex financial problem solving and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8715–8758.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Joseph E Gonzalez, Ion Stoica, and Hao Zhang. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3277–3287.

Limitations

Three limitations shape the scope of the results. First, although we now report one cross-evaluator PaySim reevaluation with a smaller secondary OpenAI model serving as the stage-3 judge, the evaluator is still usually the same LLM family that performs stage 1, and broader cross-model validation is still missing. Second, although the study now includes a stronger nearest-normal exemplar baseline, explanation-component ablations, and distributional support metrics, it still omits optimization-based or causal counterfactual methods. Third, the evidence still comes from two simulators and one real credit-risk benchmark rather than from operational financial systems with richer temporal and governance constraints.

Ethics Statement

The central ethical issue in explanation-guided counterfactual editing is not only whether the edits are valid, but how they might be used. One risk is that counterfactuals can be misused as detector-gaming recipes rather than as analytic probes. The method is therefore better suited to auditing model behavior than to generating automated recommendations for end users. A second risk is over-trust: fluent explanations and plausible edits can look persuasive even when the anomaly gate is weak. For that reason, human review should remain central to any eventual deployment scenario. Finally, two of the three datasets are simulators rather than operational financial systems, and even the real credit-risk benchmark is still a stylized tabular proxy for financial decision making. Any practical use would require domain-specific constraints, governance review, and human oversight beyond what is captured here.

Use of AI Assistants. AI tools (e.g., ChatGPT) were used for minor language editing and drafting support. All experimental design, results, and conclusions were developed and verified by the authors.

A Supplementary Protocol Materials

This appendix presents the prompt templates and protocol details used in the evaluation. For readability, it shows stable prompt skeletons and placeholders rather than a fully rendered prompt for every record.

A.1 Decision Template

The frozen decision template asks the model to classify one record as `anomalous` or `not_anomalous`, justify the decision from record-local evidence, and return structured JSON. The main decision rules are deliberately conservative: anomalies are rare, weak or ambiguous evidence should default to `not_anomalous`, and escalation requires either a hard contradiction or a coherent combination of suspicious signals.

```
You are a financial transaction risk analyst working on one
structured record at a time.

Your task:
- classify the record as anomalous or not_anomalous
- explain the decision using only the provided fields
- use anomalous only when the record contains sufficient
  evidence for escalation
- default to not_anomalous when the evidence is weak,
  ambiguous, incomplete, or underdetermined
- return only JSON with:
  predicted_label, confidence, top_features, reasons, summary

Dataset: <dataset_name>
Context: <task_description>
Decision guidance: <class_prior_hint>
Fields and meanings: <field_descriptions>
Optional calibration examples: <few_shot_examples>
Record: <record_json>
```

A.2 Editing Template

The editing template receives the original record, the structured analyst note from the decision prompt, dataset constraints, and an edit-density setting. It must return a full counterfactual record plus explicit feature-level edits. The three edit styles control how many mutable fields may change, not the order of fields in the prompt.

```
You are editing a financial transaction record so it appears
less suspicious while remaining plausible.

Your job:
- read the record and the analyst note
- produce a realistic counterfactual record that should be
  judged not_anomalous
- change only mutable fields
- preserve all unspecified fields exactly
- respect hard constraints
- return only JSON with:
  edit_style, edits, counterfactual_record, rationale

Edit style: <minimal | moderate | maximal>
Dataset: <dataset_name>
Fields and meanings: <field_descriptions>
Mutable fields: <mutable_fields>
Immutable fields: <immutable_fields>
Hard constraints: <constraints>
Original record: <record_json>
Analyst note: <explanation_json>
```

A.3 Fresh-Context Reevaluation Template

The reevaluation template mirrors the decision prompt but explicitly treats the edited record as fully standalone. It omits the original label, the original record, and the earlier explanation so that success is measured under fresh-context judgment rather than same-context consistency.

```
You are a fresh financial transaction risk analyst working on
a single standalone record.

Your task:
- classify the record as anomalous or not_anomalous
- explain the decision using only the provided fields
- do not assume any prior context, edits, or hidden labels
- return the same JSON schema as the decision prompt

Dataset: <dataset_name>
Context: <task_description>
Fields and meanings: <field_descriptions>
Record: <counterfactual_record_json>
```

B Calibration and Thresholding

Calibration uses a dedicated split, and both the prompt and the anomaly threshold are fixed before evaluation on held-out data. Each of the three main datasets uses its own frozen calibration examples and threshold. For brevity, we show only the PaySim set here as a representative concrete instance. For PaySim, calibration uses five examples: three routine transactions and two clear anomalies. The normal exemplars teach coherent balance flows for `DEBIT`, `TRANSFER`, and merchant `PAYMENT`; the anomaly exemplars teach two direct contradictions, namely a transfer whose destination balance never changes and a transfer whose destination balance increases by far more than the stated amount.

Calibration example	Purpose
Normal DEBIT	Teaches the routine two-sided debit pattern with exact balance updates.
Normal TRANSFER	Shows that a large amount alone is not sufficient for escalation.
Normal merchant PAYMENT	Shows that sparse merchant-side balance information is not a contradiction by itself.
Anomalous TRANSFER (no destination update)	Teaches a direct source-destination inconsistency.
Anomalous TRANSFER (destination jump too large)	Teaches a hard numerical contradiction on the destination side.

Table A1: Representative frozen PaySim calibration examples used for prompt calibration and threshold selection. Analogous dataset-specific calibration examples were also fixed for BankSim and Credit Default, but are omitted here for brevity.

On the PaySim calibration split, GPT-4.1 over-predicted anomalies under the raw prompt. We therefore fixed the anomaly threshold at 0.98 before evaluating on held-out data. The reported held-out results use that frozen threshold without further tuning. The same calibration protocol was applied separately to BankSim and Credit Default before their held-out evaluation.

C Oracle, Baselines, and Metrics

C.1 External Oracle

The external oracle is a random forest trained separately from the LLM pipeline for each dataset. It uses 300 trees, maximum depth 16, minimum leaf size 2, class-balanced subsampling, median imputation plus standardization for numeric fields, and most-frequent imputation plus one-hot encoding for categorical fields. Oracle validity is computed by asking whether this independently trained model predicts the edited record as normal. The oracle is therefore not another prompt-based judge. Plausibility is the mean distance to the nearest normal-record neighbors in the oracle's transformed feature space.

C.2 Programmatic Baselines

The baselines are implemented under the same mutability and constraint rules as the explanation-guided editor.

- **Random perturbation**: sample candidate replacements for mutable fields from the normal pool and greedily apply them until the oracle flips.
- **Distributional perturbation**: replace numeric fields with normal-pool medians and categorical fields with normal-pool modes, again using greedy application.
- **SHAP-guided perturbation**: rank mutable features by oracle SHAP importance and apply the same median/mode replacements in that order.
- **Nearest-normal exemplar**: map the record into the oracle's preprocessed feature space, retrieve the five nearest normal records by Euclidean k-NN from the sampled normal reference pool, and incrementally copy mutable fields from the closest exemplar while preserving the same hard constraints.

C.3 Explanation Alignment

Explanation alignment is computed from the overlap between the features named in the decision explanation and the features actually changed by the editor. Let E be the set of explanation features and C the set of changed features. In the main paper we report the Jaccard score

$$\mathrm{Jaccard}(E, C) = \frac{|E \cap C|}{|E \cup C|},$$

D Bootstrap Confidence Intervals

Statistical uncertainty is estimated with nonparametric bootstrap resampling over held-out examples. For stage 1, we resample the 10,000 held-out PaySim records and recompute precision, recall, and forwarded anomaly rate. For stage 3, we resample counterfactual rows within each method and edit setting and recompute fresh-context validity and oracle validity.

Method	Fresh-context 95% CI	Oracle-valid 95% CI
LLM minimal	[47.1, 54.3]	[91.5, 95.1]
LLM moderate	[66.2, 72.8]	[92.8, 96.0]
LLM maximal	[88.7, 92.8]	[90.0, 93.9]
Random	[11.8, 16.8]	[99.6, 100.0]
Distributional	[33.3, 39.9]	[98.0, 99.5]
SHAP-guided	[1.2, 3.2]	[100.0, 100.0]

Table A2: Bootstrap 95% confidence intervals for the main held-out PaySim metrics. The intervals remain well separated for fresh-context success.

E Label-Wise Stratification and Cross-Family Reevaluation

Table A3 separates forwarded false positives from forwarded true anomalies for the settings most relevant to the explanation-specificity question. On PaySim, the weak stage-1 gate means that smaller edit budgets work much better on forwarded false positives than on the few true anomalies that reach stage 2. On Credit Default, the split is much less skewed.

We also ran a cross-evaluator reevaluation on held-out PaySim with GPT-4.1 generating the edits and a smaller secondary OpenAI evaluator judging the edited records in stage 3. Table A4 shows that the main ordering persists even under a different evaluator.

Dataset	Setting	FP n	FP success	TP success
PaySim	Guided min	731	51.3%	31.8%
PaySim	No-exp min	732	47.8%	22.7%
PaySim	Swap min	730	48.9%	40.9%
PaySim	Guided mod	732	70.4%	40.9%
PaySim	Guided max	732	90.7%	95.5%
Credit	Guided min	656	55.6%	57.9%
Credit	No-exp min	656	32.8%	36.8%
Credit	Swap min	656	54.4%	47.4%

Table A3: Fresh-context success after separating forwarded false positives from forwarded true anomalies. The PaySim stage-1 gate is heavily false-positive dominated, so the smaller edit budgets should be interpreted with that skew in mind.

Method	Fresh success	Oracle valid
Guided min	81.9%	90.8%
No-exp min	66.9%	92.1%
Swap min	66.0%	92.9%
Guided mod	83.4%	94.3%
Guided max	92.9%	92.7%
Random	3.4%	100.0%
Distributional	0.2%	98.9%
SHAP-guided	0.0%	100.0%

Table A4: Cross-evaluator held-out PaySim reevaluation with GPT-4.1 generating edits and a smaller secondary OpenAI evaluator serving as the stage-3 judge. The absolute rates change, but the explanation-guided ordering remains.

PaySim setting	Fresh success	Oracle valid	Mean sparsity
Guided min	50.8%	93.4%	1.21
No-exp min	47.1%	93.4%	1.17
Swap min	48.5%	93.8%	1.13
Top-features-only min	42.8%	94.6%	1.30
Reasons-only min	42.0%	93.4%	1.26
Nearest exemplar	5.4%	100.0%	1.03

Table A5: Additional PaySim ablations. The explanation-component ablations show that the full explanation performs best among the matched minimal-edit variants, while the stronger nearest-normal exemplar baseline remains much easier to validate under the tabular reference model than under fresh LLM reevaluation.

F Stage-1 Gating Across Models

Table A6 consolidates the stage-1 gate diagnostics for the main GPT-4.1 runs and the later open-model transfer checks. The GPT-4.1 rows for PaySim, BankSim, and Credit Default use the held-out aggregate reported in the main paper. The open-model rows are single-split exploratory transfer runs. For reference, we also include the independently trained random-forest oracle on its own held-out test split for the three main datasets.

G Protocol Safeguards and Reproducibility

The main evaluation protocol treats leakage control as part of the scientific claim rather than as an implementation detail. Concretely, the evaluation satisfies the following conditions.

- Ground-truth labels are never shown to the LLM.
- PaySim simulator alert metadata such as `isFlaggedFraud` is excluded from prompts and from the oracle feature view.
- Prompt calibration uses only the dedicated calibration split.
- The prompt, calibration examples, mutable-field list, and anomaly threshold are fixed before evaluation.
- The reevaluation prompt treats the edited record as fully standalone and does not expose prior pipeline context.
- All reported main results use only the no-leakage configuration described above.

Dataset	Model	Prompt	n	Thr.	Fwd	TP	FP	FN	TN	Prec.	Rec.	F1	AUC
PaySim	GPT-4.1	default-heldout	10,000	0.98	754	22	732	78	9,168	2.9	22.0	5.2	–
PaySim	Qwen-14B	default	5,000	0.00	3,955	49	3,906	1	1,044	1.2	98.0	2.4	–
PaySim	Mistral-8B	default	5,000	0.98	1	1	0	49	4,950	100.0	2.0	3.9	–
PaySim	RF oracle	test-split	10,229	0.50	205	199	6	29	9,995	97.1	87.3	91.9	99.4
BankSim	GPT-4.1	default-heldout	10,000	0.00	26	25	1	75	9,899	96.2	25.0	39.7	–
BankSim	Qwen-14B	default	5,000	0.00	7	6	1	44	4,949	85.7	12.0	21.1	–
BankSim	Mistral-8B	default	5,000	0.00	68	14	54	36	4,896	20.6	28.0	23.7	–
BankSim	RF oracle	test-split	30,240	0.50	2,320	1,335	985	105	27,815	57.5	92.7	71.0	99.3
Credit Default	GPT-4.1	default-heldout	10,000	0.00	694	38	656	62	9,244	5.5	38.0	9.6	–
Credit Default	Qwen-14B	default	5,000	0.00	670	15	655	35	4,295	2.2	30.0	4.2	–
Credit Default	Mistral-8B	default	5,000	0.00	3,448	36	3,412	14	1,538	1.0	72.0	2.1	–
Credit Default	RF oracle	test-split	6,000	0.50	1,133	676	457	651	4,216	59.7	50.9	55.0	79.3

Table A6: Stage-1 gating diagnostics across the main GPT-4.1 pipeline and the later model-transfer runs. "Fwd" is the number of records forwarded into stage 2 after applying the stage-1 gate. Precision, recall, F1, and AUC are percentages. The simulator and credit rows use either the held-out GPT aggregate or the single-split open-model stress-test runs described in the text. The "RF oracle" rows report the independently trained random forest on its own held-out test split using the standard 0.5 anomaly threshold.

H Representative GPT-4.1 Explanation Traces

Table A7 gives a small set of illustrative GPT-4.1 traces drawn from the completed explanation-guided runs. These examples are not a benchmark summary. Instead, they show how the model framed stage-1 escalation, which fields it changed in the counterfactual, and how the fresh stage-3 judgment responded. The table includes three representative successes from the main datasets and one representative failure case. To keep the rows readable, each example shows only a short record snippet rather than the full original input.

Table A7: Representative GPT-4.1 explanation traces from the completed runs. Each row shows an original record snippet, the stage-1 prediction and summary, the edited counterfactual fields, and the fresh stage-3 judgment. These rows are illustrative traces rather than a benchmark summary, and the edited fields are shown explicitly as `old` → `new` changes.

Dataset / row	Original record snippet	Stage-1 decision	Counterfactual / edited fields	Stage-3 decision
PaySim Record 198583 Gold: normal Edit: minimal Stage 3: normal Representative success	`type`: TRANSFER `amount`: 325920.81 `oldbalanceOrg`: 0 `newbalanceOrig`: 0 `newbalanceDest`: 1971972.73	**Prediction**: anomalous (0.98) This record is anomalous because the origin account has a zero balance both before and after the transfer, yet a large amount is transferred and credited to the destination. This is a direct contradiction in the balance flow and is not possible in a valid transaction. Top features: `type`, `amount`, `oldbalanceOrg`, `newbalanceOrig`	**`oldbalanceOrg`: 0 → 325920.81**	**Prediction**: not_anomalous TRANSFER is a standard transaction type and does not by itself indicate anomaly. Top features: `type`, `amount`, `oldbalanceOrg`, `newbalanceOrig`
BankSim Record 27399 Gold: anomalous Edit: moderate Stage 3: normal Representative success	`category`: es_otherservices `amount`: 736.28 `same_zipcode`: 1	**Prediction**: anomalous (0.91) This record is anomalous because it combines a very large transaction amount with a broad, non-specific service category, creating a jointly suspicious pattern that warrants escalation despite the shared zipcode. Top features: `amount`, `category`, `same_zipcode`	**`category`: es_otherservices → es_barsandrestaurants** **`amount`: 736.28 → 68.5**	**Prediction**: not_anomalous Bars and restaurants are a common consumer spending category and do not raise suspicion by themselves. Top features: `category`, `amount`, `same_zipcode`
UCI Credit Default Record 2510 Gold: anomalous Edit: minimal Stage 3: normal Representative success	`PAY_0`: 2 `PAY_2`: 2 `BILL_AMT1`: 31202 `PAY_AMT2`: 0	**Prediction**: anomalous (0.93) This record is anomalous because it shows a sustained two-month delinquency, high balances at or above the credit limit, and missed payments, all of which jointly indicate significant repayment stress and warrant escalation. Top features: `PAY_0`, `PAY_2`, `BILL_AMT1`, `PAY_AMT2`	**`PAY_0`: 2 → 0**	**Prediction**: not_anomalous The most recent repayment status is current (0), indicating no present delinquency. Top features: `PAY_0`, `PAY_2`, `BILL_AMT1`, `PAY_AMT2`
UCI Credit Default Record 112 Gold: normal Edit: minimal Stage 3: anomalous Representative failure	`PAY_0`: 2 `PAY_2`: 2 `PAY_6`: 3 `BILL_AMT1`: 135673 `PAY_AMT2`: 0	**Prediction**: anomalous (0.95) This record is anomalous because it shows a persistent multi-month delinquency pattern, large outstanding balances, and insufficient repayment activity, all of which jointly indicate a high risk of default and warrant escalation. Top features: `PAY_0`, `PAY_2`, `PAY_6`, `BILL_AMT1`	**`PAY_0`: 2 → 0**	**Prediction**: anomalous The most recent repayment status is current (0), but this follows a sustained period of delinquency, suggesting a recent cure rather than a stable pattern. Top features: `PAY_0`, `PAY_2`, `PAY_6`, `BILL_AMT1`