ENRICHING SCARCE ROOT-CAUSE SUPERVISION FOR FAILURE ATTRIBUTION VIA SOURCE-CONDITIONED SUPERVISION REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Failure attribution in LLM-based multi-agent systems aims to identify the decisive root-cause step within a long execution trajectory, yet each failed trajectory typically contains only one such step among many non-root-cause ones, resulting in highly imbalanced supervision. Latent-space synthesis provides a natural way to expand supervision around scarce root-cause examples by constructing source-conditioned latent neighborhoods, but naive expansion introduces two challenges: source-level training influence can grow with synthesis multiplicity, while individual synthetic samples may differ substantially in positive-support compatibility and redundancy. We introduce Source-conditioned Supervision Refinement (SSR), which fixes the total synthetic supervision weight contributed by each source and softly adjusts sample-level supervision according to positive-support compatibility and source-specific non-redundancy. This preserves broad local coverage while emphasizing better-supported and less redundant synthetic samples. SSR operates only during training; inference uses deterministic latent encoding and a lightweight attribution detector without additional generation, sampling, or trajectory replay. Experiments on three failure-attribution benchmarks show that SSR consistently outperforms matched representation and synthesis baselines.

1 INTRODUCTION

LLM-based multi-agent systems increasingly solve complex tasks through coordinated agent actions, inter-agent communication, and external tool use (Wu et al., 2023; Hong et al., 2024; Qian et al., 2024; Yao et al., 2022; Fourney et al., 2024; Li et al., 2023). Despite their growing capabilities, these systems can still fail during long-horizon execution, making it important to identify the decisive step responsible for the failure, a task known as failure attribution (Zhang et al., 2025b; Deshpande et al., 2025; Liu et al., 2026; Hutter & Pradel, 2026). This problem is particularly challenging in multi-agent trajectories: an early erroneous action, message, or tool interaction can propagate through subsequent coordination, while later behaviors and partial recovery may obscure the original cause and make it difficult to distinguish the root cause from its downstream consequences (Yu et al., 2025; West et al., 2025; In et al., 2026; Jia et al., 2026). Moreover, because each failed trajectory typically provides only a single root-cause step among many execution steps, step-level attribution naturally operates under highly sparse positive supervision.

A natural way to alleviate the resulting scarcity of root-cause supervision is latent-space synthesis, which generates controlled variations around observed root-cause representations without constructing complete failure trajectories in text space (DeVries & Taylor, 2017; Verma et al., 2019; Dablain et al., 2022; Mondal et al., 2023; Tao et al., 2025). However, using synthetic positives effectively for rare-positive failure attribution is non-trivial. First, increasing synthesis multiplicity not only broadens the latent neighborhood around a root-cause source but also increases its aggregate synthetic weight under naive unit weighting. Second, synthetic root-cause samples are not equally informative for supervision: some may drift away from the local support of real root-cause samples, while others may provide little additional coverage because they are highly redundant with samples generated from the same source. As illustrated in Figure 1, naively treating all synthetic samples as equivalent supervision can amplify source-level influence and introduce noisy or redundant signals that interfere

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.

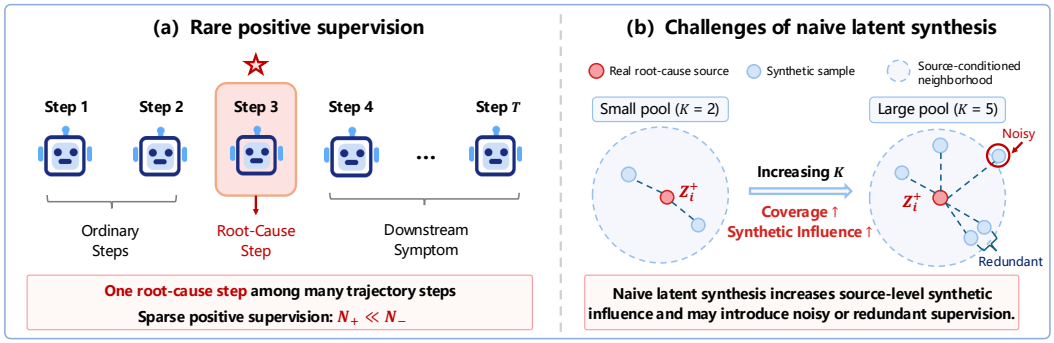


Figure 1: **Rare-positive supervision and challenges of naive latent synthesis in failure attribution.** (a) Each failed trajectory contains one root-cause step but many non-root-cause steps, yielding highly sparse positive supervision. (b) Naive latent synthesis expands the source-conditioned latent neighborhood, but greater synthesis multiplicity can amplify source-level synthetic influence, while synthetic root-cause samples may vary in positive-support compatibility and redundancy.

with detector learning. The resulting synthetic supervision should therefore be controlled at the source level and selectively refined across samples.

In this work, we investigate an important yet underexplored research question:

How can we enrich scarce root-cause supervision while refining synthetic supervision at both the source and sample levels?

To address this, we introduce **S**ource-conditioned **S**upervision **R**efinement (SSR), a framework for enriching scarce root-cause supervision and refining the resulting synthetic supervision. SSR uses a variational autoencoder (VAE) to learn a structured latent space that supports localized perturbations around each observed root-cause source. These perturbations generate synthetic root-cause samples that form a source-conditioned latent neighborhood, providing additional supervision without requiring trajectory-level generation. Rather than treating all synthetic samples as equally reliable, SSR fixes their total synthetic supervision weight at the source level and softly refines their individual supervision using positive-support compatibility and source-specific non-redundancy.

Positive-support compatibility measures whether a synthetic sample remains supported by real root-cause examples relative to nearby negatives, while source-specific non-redundancy measures the absence of a close substitute among candidates from the same source. Together, they distribute a fixed amount of source-level synthetic supervision across samples, yielding source-controlled, sample-adaptive supervision for detector learning.

Once trained, SSR performs failure attribution through deterministic latent encoding followed by a lightweight attribution detector, requiring no sampling, additional reasoning, or trajectory replay at test time. Across three failure-attribution benchmarks, SSR consistently outperforms matched representation and synthesis baselines while retaining lightweight single-pass inference.

Our key contributions are summarized as follows:

- We highlight two limitations of naive latent synthesis for rare-positive failure attribution: source-level synthetic influence grows with synthesis multiplicity, while synthetic samples vary in positive-support compatibility and redundancy.
- We propose SSR, which constructs source-conditioned latent neighborhoods and refines their synthetic supervision through source-level control and utility-guided sample weighting.
- We provide empirical results showing consistent gains across multiple benchmarks, together with theoretical analyses of the refinement mechanism, while retaining lightweight single-pass inference.

2 Related Work

Failure attribution in LLM-based agent systems has attracted increasing attention in recent years (Zhang et al., 2025b; Deshpande et al., 2025; Zhang et al., 2025a; Kong et al., 2025; In et al., 2026; Jia et al., 2026). One line of work performs attribution through LLM-based diagnosis, including prompting, abductive reasoning, and judge models (Zhang et al., 2025b; West et al., 2025; Zhu et al.,

2026a; Banerjee et al., 2025; Yu et al., 2025). Another line estimates responsibility from interaction structures, execution replay, or error propagation (Ge et al., 2025; Ma et al., 2025). Recent work further organizes failure attribution through hierarchical scope reduction and structured trajectory modeling, including two-stage scope-before-localization frameworks and adaptive influence-graph construction (Sun et al., 2026; Bakish et al., 2026). Meanwhile, recent approaches also train attribution models directly from execution traces or step representations (Zhu et al., 2026b; Li et al., 2026).

While effective for failure diagnosis and localization, existing methods are generally not designed for the rare-positive regime of learned step-level attribution. Our work is complementary to existing diagnostic, structural, and representation-based attribution methods: rather than introducing an additional diagnostic mechanism, SSR enriches and refines the scarce root-cause supervision available to the attribution detector.

Latent-space synthesis and synthetic sample control have been widely studied to enrich limited or imbalanced supervision in learned feature or latent spaces. One line of work synthesizes minority examples through interpolation or perturbation over learned representations (Dablain et al., 2022; Mondal et al., 2023). Another line improves synthetic-sample quality through structured latent spaces, density estimation, or likelihood-based filtering (Hong et al., 2025; Tao et al., 2025), while other work promotes diversity to broaden minority-class coverage (Yang et al., 2023).

These approaches primarily focus on how synthetic samples are generated, assessed, or diversified. In contrast, we focus on how the synthetic supervision induced by a source-conditioned latent neighborhood should contribute to downstream attribution learning. SSR refines this supervision through source-level control and sample-level utility-guided weighting.

3 Problem Setup

We formalize multi-agent execution trajectories, step-level root-cause attribution, and the rare-positive supervision regime considered in this work.

Multi-agent execution trajectory. We represent a failed execution produced by an LLM-based multi-agent system as an ordered sequence

$$\tau = (x_1, x_2, \ldots, x_T), \tag{1}$$

where each step x_t may correspond to an agent action, inter-agent message, tool call, observation, or environment response, together with the identity of the acting agent when applicable.

Root-cause attribution. Given a failed trajectory τ, failure attribution aims to identify the decisive step responsible for the task failure rather than downstream steps that merely reflect its consequences. Let $y_t \in \{0, 1\}$ indicate whether x_t is the root cause. Following the benchmark annotations considered in this work, each failed trajectory contains a single labeled root-cause step,

$$y_{t^*} = 1, \qquad \sum_{t=1}^{T} y_t = 1, \tag{2}$$

where t^* denotes its ground-truth position. The goal is to learn an attribution model that assigns a root-cause likelihood to each step and identifies the highest-scoring step as the predicted root cause.

Key challenge. Latent-space synthesis provides additional supervision around scarce root-cause examples, but it also introduces substantial variation in source-level influence and sample quality. As a result, treating synthetic root-cause samples uniformly can be unreliable: their aggregate influence may grow with synthesis multiplicity, while some samples may be weakly supported or highly redundant. Our approach addresses this by refining source-conditioned synthetic supervision through source-level control and sample-level utility-guided weighting, which we introduce next.

4 Method

We propose SSR, a source-conditioned supervision refinement framework for enriching scarce root-cause supervision. SSR first learns a structured latent representation of trajectory steps and

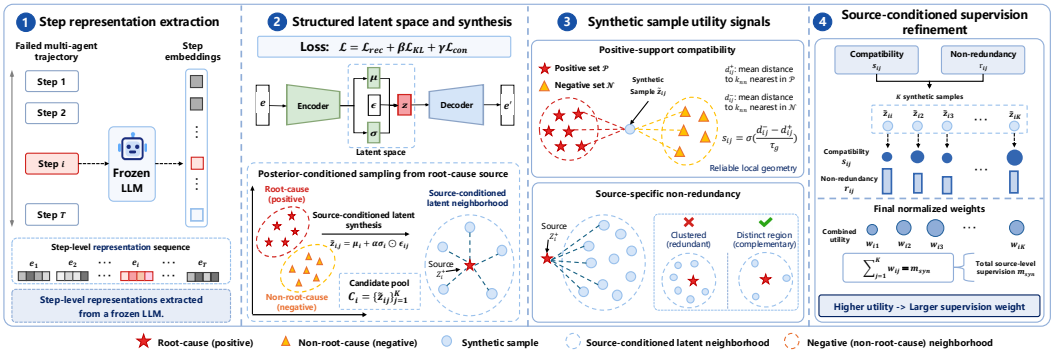


Figure 2: **Overview of SSR for rare-positive failure attribution in LLM-based multi-agent systems.** SSR first constructs source-conditioned latent neighborhoods around observed root-cause sources through latent synthesis, and then refines the resulting synthetic supervision using source-level control and utility-guided sample weighting. This enables broader root-cause supervision while fixing the total synthetic supervision weight of each source and emphasizing well-supported, non-redundant synthetic samples for downstream failure attribution.

synthesizes local variations around each observed root-cause source. Rather than treating these synthetic samples as equally reliable supervision, SSR further refines how they contribute to attribution learning through source-level control and sample-level utility-guided weighting.

The key motivation is that expanding a synthetic neighborhood does not necessarily yield better supervision: increasing synthesis multiplicity can amplify source-level synthetic influence, while individual synthetic root-cause samples may vary substantially in positive-support compatibility and redundancy. SSR therefore separates neighborhood expansion from supervision refinement, fixing the total synthetic supervision weight at the source level while adapting supervision across samples within each source-conditioned neighborhood.

Overview. Given a failed trajectory τ, SSR first learns a probabilistic latent representation of contextual step embeddings and synthesizes local variations around each observed root-cause source in the learned latent space. These synthetic root-cause samples form a source-conditioned latent neighborhood. SSR then controls the aggregate synthetic supervision contributed by each source and distributes this supervision across samples according to two complementary signals: *positive-support compatibility*, which measures consistency with real root-cause support, and *source-specific non-redundancy*, which measures the absence of a close same-source substitute. The resulting weighted samples are combined with real trajectory steps for attribution learning. Our framework consists of two integral components:

- *Source-conditioned neighborhood construction.* We learn a probabilistic latent representation of trajectory steps and construct a local synthetic neighborhood around each observed root-cause source through controlled perturbations.
- *Source-conditioned supervision refinement.* We control aggregate synthetic supervision at the source level and softly refine sample-level supervision according to positive-support compatibility and source-specific non-redundancy.

The language model remains fixed throughout, and all latent synthesis and supervision refinement are performed only during training. Figure 2 summarizes the framework.

4.1 Structured Latent Representation and Synthesis

Structured latent representation. For each step x_t in a failed trajectory τ, we extract its contextual LLM embedding as

$$\mathbf{e}_t = \mathrm{LLM}_{\mathrm{embed}}(x_t), \tag{3}$$

where $\mathrm{LLM}_{\mathrm{embed}}$ denotes the contextual embedding extraction procedure applied to step x_t using a frozen language model, with the hidden-state representation taken from the penultimate layer. These embeddings capture the contextual semantics of individual execution steps and serve as the starting point for VAE-based latent modeling and synthesis.

Rather than synthesizing directly in the high-dimensional LLM embedding space, we leverage a variational autoencoder (VAE) (Kingma & Welling, 2013) to learn a smooth and structured latent

representation of contextual step embeddings. By mapping each step embedding to a posterior distribution rather than a fixed point, the VAE provides a natural basis for localized perturbations around observed root-cause sources. Its encoder defines

$$q_\phi(\mathbf{z} \mid \mathbf{e}) = \mathcal{N}\big(\mathbf{z}; \mu_\phi(\mathbf{e}), \mathrm{diag}(\sigma_\phi^2(\mathbf{e}))\big), \qquad \mathbf{z} = \mu_\phi(\mathbf{e}) + \sigma_\phi(\mathbf{e}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

and a decoder reconstructs the original step embedding. The decoder is used only during latent representation learning for reconstruction; subsequent synthesis and attribution learning operate directly in the latent space.

To encourage attribution-discriminative latent structure, we further regularize the VAE with a contrastive objective that promotes compactness among positive representations and among negative representations while increasing separation between the two classes. The overall objective is

$$\mathcal{L}_{\text{latent}} = \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{KL}} + \gamma \mathcal{L}_{\text{con}}, \tag{5}$$

where the reconstruction and KL terms preserve representation fidelity and regularize the posterior, while the contrastive term explicitly promotes positive-class compactness, negative-class compactness, and positive–negative separation. For real trajectory steps, we use the deterministic posterior mean $\mathbf{z}_t = \mu_\phi(\mathbf{e}_t)$.

After latent representation learning, the VAE encoder is frozen, so subsequent synthesis, supervision refinement, and detector training operate in a fixed latent space. The complete training objective and implementation details are provided in Appendix A.

Source-conditioned synthesis. Once the VAE has been trained, we synthesize additional root-cause samples by performing controlled perturbations around observed root-cause representations in the learned latent space. For the i-th observed root-cause source, we generate

$$\tilde{\mathbf{z}}_{ij} = \mu_\phi(\mathbf{e}_i^+) + \alpha \sigma_\phi(\mathbf{e}_i^+) \odot \boldsymbol{\epsilon}_{ij}, \qquad \boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $j = 1, \ldots, K$. The resulting synthetic root-cause samples form the candidate collection $\mathcal{C}_i = (\tilde{\mathbf{z}}_{ij})_{j=1}^K$, which defines a source-conditioned latent neighborhood around source i. Here, K controls synthesis multiplicity, while α controls how broadly the local neighborhood is explored.

4.2 Source-conditioned Supervision Refinement

Why source-conditioned refinement? Synthetic samples generated from the same root-cause source are local variations of a shared observed positive rather than independent sources of supervision. As a result, increasing synthesis multiplicity should expand local coverage without proportionally amplifying source-level influence, while supervision within each neighborhood should account for differences in sample reliability and redundancy. Our refinement therefore fixes the total synthetic supervision weight at the source level and redistributes supervision across samples according to their local utility.

Source-level supervision control. Let $w_{ij} \geq 0$ denote the supervision weight assigned to synthetic sample $\tilde{\mathbf{z}}_{ij}$. Treating every synthesized sample as an independent supervision unit makes the aggregate contribution of a root-cause source grow with synthesis multiplicity. We instead constrain the aggregate synthetic weight of source i as $\sum_{j=1}^K w_{ij} = m_{\text{syn}}$, where $m_{\text{syn}} > 0$ controls the total synthetic supervision contributed by each source and is independent of K. This source-level normalization decouples synthesis multiplicity from the total synthetic supervision weight available for sample-level refinement.

Positive-support compatibility. Let $\mathcal{P} = \{\mathbf{z}_n : y_n = 1\}$ and $\mathcal{N} = \{\mathbf{z}_n : y_n = 0\}$ denote the real positive and negative latent representations. For candidate $\tilde{\mathbf{z}}_{ij}$, let d_{ij}^+ and d_{ij}^- denote the mean cosine distances to its k_{nn} nearest positive and negative reference representations, respectively. We define

$$g_{ij} = d_{ij}^- - d_{ij}^+, \qquad s_{ij} = \mathrm{sigmoid}\left(\frac{g_{ij}}{\tau_g}\right), \tag{7}$$

where $\tau_g > 0$ is the geometry temperature and s_{ij} is the *positive-support compatibility*. A larger s_{ij} indicates stronger relative support from real root-cause examples. We use s_{ij} as a compatibility score rather than a calibrated class probability.

Source-specific non-redundancy. Compatibility alone does not distinguish samples that repeatedly represent the same local region. We therefore define

$$\Delta_{ij}^{\mathrm{nr}} = 1 - \max_{v \neq j} \mathrm{sim}\big(\tilde{\mathbf{z}}_{ij}, \tilde{\mathbf{z}}_{iv}\big), \qquad r_{ij} = \exp\left(\frac{\Delta_{ij}^{\mathrm{nr}}}{\tau_r}\right), \tag{8}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity and $\tau_r > 0$ is the non-redundancy temperature. A larger $\Delta_{ij}^{\mathrm{nr}}$ indicates that $\tilde{\mathbf{z}}_{ij}$ has no close substitute within the same source-conditioned candidate set. For $K = 1$, we set $r_{i1} = 1$, since no within-source redundancy comparison is required. Its equivalent facility-style interpretation for $K \geq 2$ is provided in Appendix I.

Utility-guided soft refinement. We combine the two signals and distribute the source-level supervision proportionally to sample utility:

$$u_{ij} = r_{ij} s_{ij}^{q}, \qquad w_{ij} = m_{\mathrm{syn}} \frac{u_{ij} + \varepsilon}{\sum_{a=1}^{K} (u_{ia} + \varepsilon)}, \tag{9}$$

where $q \geq 0$ controls the contribution of positive-support compatibility and $\varepsilon > 0$ ensures numerical stability. By construction, $\sum_{j=1}^{K} w_{ij} = m_{\mathrm{syn}}$.

This weighting emphasizes samples that are well supported by real root-cause examples and less redundant within the source-conditioned neighborhood, while softly downweighting less reliable or highly redundant ones. Sample utility determines relative supervision within each neighborhood, whereas source-level normalization fixes its total synthetic supervision weight.

Attribution learning. The refined synthetic root-cause samples are combined with real trajectory steps to train the attribution detector. Real samples retain unit weight, whereas each synthetic root-cause sample $\tilde{\mathbf{z}}_{ij}$ contributes according to its refined supervision weight w_{ij}. Given detector p_ω, we optimize the weighted binary cross-entropy objective

$$\mathcal{L}_{\mathrm{attr}} = \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} \eta_n \, \ell_{\mathrm{BCE}}\big(p_\omega(\mathbf{z}_n), y_n\big), \tag{10}$$

where $\eta_n = 1$ for real trajectory samples and $\eta_n = w_{ij}$ for synthetic root-cause samples. Thus, synthetic samples enrich positive supervision while their contribution to detector optimization remains explicitly controlled.

Mathematical interpretation. Source-level normalization directly controls the explicitly weighted synthetic-loss contribution of each source. Let $\mathbf{g}_{ij}$ denote the gradient of the BCE loss induced by synthetic sample $\tilde{\mathbf{z}}_{ij}$ with respect to the detector parameters under a fixed training realization.

Proposition 1 (Source-level control of weighted synthetic gradients)**.** *Suppose the synthetic-sample gradients are uniformly bounded as $\|\mathbf{g}_{ij}\|_2 \leq G$ for a common constant G independent of K. For non-negative weights satisfying $\sum_{j=1}^{K} w_{ij} = m_{\mathrm{syn}}$,*

$$\left\| \sum_{j=1}^{K} w_{ij} \mathbf{g}_{ij} \right\|_2 \leq m_{\mathrm{syn}} G, \tag{11}$$

independently of synthesis multiplicity K. Under unit weighting, the corresponding worst-case bound is KG.

Proposition 1 shows that the worst-case aggregate weighted gradient contribution of each source remains bounded independently of synthesis multiplicity K under source-level normalization. Additional results in Appendix I establish a bounded-error interpretation of positive-support compatibility and an exact same-source nearest-neighbor interpretation of source-specific non-redundancy, providing structural justification for the two signals used in sample-level supervision refinement.

4.3 Test-time Attribution

At test time, given a failed trajectory $\overline{\tau} = (\overline{x}_1, \ldots, \overline{x}_T)$, we extract contextual step embeddings $\overline{\mathbf{e}}_t = \mathrm{LLM}_{\mathrm{embed}}(\overline{x}_t)$ and obtain deterministic latent representations $\overline{\mathbf{z}}_t = \mu_\phi(\overline{\mathbf{e}}_t)$, which are scored

Table 1: **Comparison with controlled mechanism and task-specific failure attribution baselines.** All values are percentages (trajectory-level Top-1 Acc.) on different failure attribution datasets. All results are reported as mean ± standard deviation over five random seeds. The best results are highlighted in **bold**.

Method	Who&When-AG	Who&When-HC	TraceElephant	CORRECT-Error
Mechanism baselines				
Raw Embedding	46.03 ± 1.94	8.97 ± 3.93	8.55 ± 4.15	41.62 ± 3.75
Latent Mean	45.71 ± 2.07	13.10 ± 2.89	6.36 ± 1.11	36.96 ± 1.99
Latent Sample-1	29.52 ± 3.65	8.28 ± 1.89	6.91 ± 1.52	30.82 ± 1.73
Naive Latent Synthesis	47.30 ± 2.84	10.34 ± 3.45	7.45 ± 2.44	35.09 ± 2.45
Raw-Space Augmentation	48.57 ± 1.42	12.41 ± 3.08	11.27 ± 3.56	38.33 ± 3.53
Task baselines				
LLM-as-Judge (Zhang et al., 2025b)	11.43 ± 3.95	6.21 ± 2.89	7.45 ± 1.35	15.02 ± 0.93
ECHO (Banerjee et al., 2025)	20.00 ± 4.14	12.41 ± 3.93	14.73 ± 1.63	20.41 ± 0.91
CORRECT (Yu et al., 2025)	19.37 ± 5.19	13.10 ± 1.54	15.09 ± 1.52	23.40 ± 0.80
A2P (West et al., 2025)	35.87 ± 1.42	17.93 ± 5.11	8.18 ± 1.29	25.21 ± 0.57
MASPrism (Liu et al., 2026)	31.11 ± 5.22	16.55 ± 1.54	6.00 ± 1.99	21.15 ± 0.55
AgenTracer (Zhang et al., 2025a)	33.02 ± 5.31	18.62 ± 4.63	14.00 ± 1.65	14.45 ± 0.13
StepFinder (Zhu et al., 2026b)	20.32 ± 3.44	12.41 ± 3.08	14.73 ± 1.63	19.93 ± 1.31
SSR (Ours)	**55.56 ± 1.94**	**21.38 ± 3.78**	**15.64 ± 1.49**	**47.96 ± 1.78**

by the attribution detector as $\overline{\rho}_t = p_\omega(\overline{\mathbf{z}}_t)$ to predict $\hat{t} = \arg\max_t \overline{\rho}_t$. Importantly, SSR does *not* require any synthesis or supervision refinement at test time; all synthetic perturbations and sample weighting are used only during training.

5 Experiments

In this section, we present comprehensive empirical evidence to validate the effectiveness of SSR on diverse failure attribution tasks. Section 5.1 details the setup, and Sections 5.2–5.3 provide results and detailed analyses.

5.1 Setup

Datasets and models. We evaluate SSR on three representative failure attribution benchmarks covering diverse multi-agent tasks and trajectory construction settings. We use Who&When (Zhang et al., 2025b), which contains an algorithm-generated subset (Who&When-AG) with 126 failure trajectories and a hand-crafted subset (Who&When-HC) with 58 longer trajectories. We further use TraceElephant (Chen et al., 2026), which contains 220 fully observable multi-agent trajectories, and CORRECT-Error (Yu et al., 2025), which contains 2,226 validated error-injected trajectories spanning diverse multi-agent applications. For all datasets, we reserve 50% of the available trajectories for testing, 10% for validation, and the remaining 40% for training. Additional experiments under different split ratios are provided in Appendix C.

We evaluate our method using two LLM families: Qwen3-8B/14B (Yang et al., 2025) and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), which provide accessible internal representations suitable for contextual step embedding extraction. All models use pretrained weights and remain frozen during representation extraction. Additional results on broader model coverage are provided in Appendix F.

Baselines. We first compare SSR with five controlled mechanism variants: Raw Embedding (original LLM embeddings $\mathbf{e}_t$), Latent Mean (deterministic latent representations $\mathbf{z}_t = \mu_\phi(\mathbf{e}_t)$), Latent Sample-1 (a single stochastic latent sample $\tilde{\mathbf{z}}_{i1}$), Naive Latent Synthesis (multiple latent synthetic samples $\{\tilde{\mathbf{z}}_{ij}\}_{j=1}^{K}$), and Raw-Space Augmentation (multiple augmented samples in the original embedding space). These variants isolate the effects of latent representation, stochastic sampling, synthetic expansion, and supervision refinement. We then compare SSR with a comprehensive collection of failure attribution baselines, including LLM-based diagnosis (LLM-as-Judge (Zhang et al., 2025b)), LLM fine-tuning-based attribution (AgenTracer (Zhang et al., 2025a)), lightweight step-level prediction (StepFinder (Zhu et al., 2026b)), and recent attribution methods including ECHO (Banerjee et al., 2025), CORRECT (Yu et al., 2025), A2P (West et al., 2025), and MASPrism (Liu et al., 2026). To ensure a fair comparison, all applicable baselines are evaluated on identical test data using the default configurations described in their respective papers. More baseline details are provided in Appendix A.

Evaluation. Because each failed trajectory is associated with a single annotated root-cause step, we report Top-1 root-cause identification accuracy (Top-1 Acc.), defined as the percentage of trajectories

Table 2: **Effect of different synthetic supervision refinement strategies.** All values are percentages (trajectory-level Top-1 Acc.) over five random seeds. The best results are highlighted in **bold**.

Method	Who&When-AG	Who&When-HC	TraceElephant
Uniform Weighting	46.03 ± 1.59	10.34 ± 4.22	6.36 ± 1.11
Shuffled Weighting	49.21 ± 4.05	9.66 ± 3.78	7.27 ± 1.44
Positive-Support-Only Weighting	48.89 ± 2.07	12.41 ± 3.93	10.00 ± 2.80
Non-Redundancy-Only Weighting	49.52 ± 2.07	11.72 ± 1.89	10.00 ± 3.02
Positive-Support + Non-Redundancy (Hard)	51.43 ± 4.84	14.48 ± 2.89	10.36 ± 2.54
SSR (Ours)	**55.56 ± 1.94**	**21.38 ± 3.78**	**15.64 ± 1.49**

for which the highest-scoring predicted step exactly matches the annotated cause. Additional results under AUPRC and AUROC are provided in Appendix G.

Implementation Details. The VAE and lightweight attribution detector are trained on the training partition, while hyperparameters are selected using the validation split. The layer for training SSR, latent dimension, perturbation scale, synthesis multiplicity K, neighborhood size k_{nn}, source-level supervision strength m_{syn}, and positive-support modulation exponent q are selected using the validation split and analyzed in Section 5.3 and Appendix E. Detailed architectures and optimization settings are provided in Appendix A.

5.2 Main Results

Effect of source-conditioned supervision refinement. Table 1 first compares SSR with controlled mechanism variants. Across datasets, SSR consistently outperforms all variants, indicating that the gains do not arise from representation learning or synthetic expansion alone. On Who&When-AG, Top-1 Acc. improves from 46.03% with Raw Embedding and 47.30% with Naive Latent Synthesis to 55.56%. Similar trends hold on the other datasets, indicating that the gains arise from refining synthetic supervision rather than latent expansion alone.

Comparison with competitive failure attribution baselines. Table 1 further compares SSR against a broad suite of failure attribution baselines, including LLM-based diagnosis, LLM fine-tuning-based attribution, lightweight step-level prediction, and recent task-specific methods. SSR achieves the strongest performance across all four evaluation settings. On Who&When-AG, SSR reaches 55.56% Top-1 Acc., substantially outperforming the strongest task baseline at 35.87%. Similar gains hold on Who&When-HC and CORRECT-Error, while SSR also achieves the highest mean Top-1 Acc. on TraceElephant. Beyond accuracy, SSR remains computationally lightweight: unlike LLM-based diagnosis methods that require additional generations at inference or LLM fine-tuning-based approaches such as AgenTracer that incur substantially heavier training and rollout costs, SSR directly produces step-level attribution scores from a single observed trajectory. Additional results on broader model coverage and cross-layer generalization are provided in Appendices F and E, and computational cost is analyzed in Appendix D.

5.3 A Closer Look at SSR

In this section, we conduct a series of in-depth analyses to understand SSR. Additional ablations are provided in Appendix E.

Ablation on different supervision refinement strategies. Table 2 systematically compares alternative strategies for allocating synthetic supervision within each root-cause source. In addition to our utility-guided soft refinement, we consider uniform allocation and shuffled allocation. For shuffled allocation, we preserve the learned sample weights but randomly permute their correspondence with synthetic samples. We further consider positive-support-only weighting, non-redundancy-only weighting, and hard positive-support–non-redundancy selection.

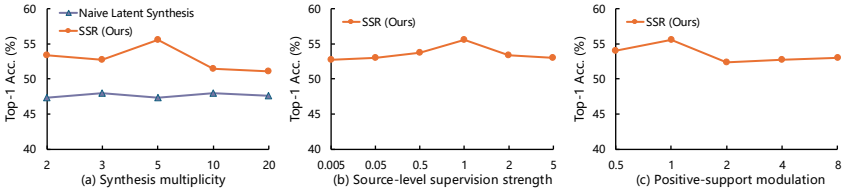


Figure 3: (a) Effect of synthesis multiplicity K, (b) effect of source-level supervision strength m_{syn}, and (c) effect of positive-support modulation exponent q. All results are averaged over five random seeds on Who&When-AG using Qwen3-8B.

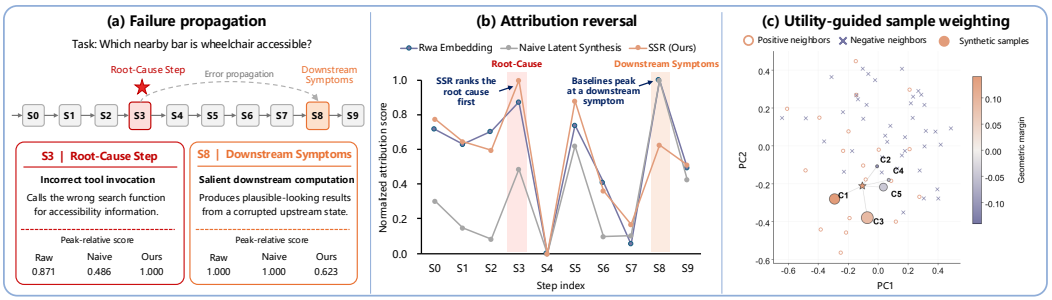


Figure 4: **Qualitative results.** (a) Root-cause attribution scores of SSR, the raw embedding detector, and naive latent synthesis on a failure trajectory; larger scores indicate higher root-cause likelihood. (b) Step-wise attribution scores over the complete trajectory in (a). (c) Latent-space visualization of source-conditioned synthetic root-cause samples and their refined supervision weights, with marker size denoting sample weight. We use Qwen3-8B for embedding extraction and a trajectory from Who&When-AG.

Overall, SSR achieves the strongest performance across the considered supervision refinement strategies. Unlike uniform or shuffled allocation, which either ignore sample utility or break the correspondence between utility and assigned supervision, our soft refinement jointly considers positive-support compatibility and source-specific non-redundancy. Using either signal alone improves over uniform allocation, indicating that both capture useful information for synthetic supervision. Combining both signals yields the strongest refinement, while hard selection performs worse than soft weighting, supporting the use of continuous supervision refinement over discrete sample removal.

How does synthesis multiplicity affect performance? Figure 3(a) shows that SSR consistently outperforms Naive Latent Synthesis across the tested values of K, with the best performance at a moderate synthesis multiplicity. Small K provides fewer local variations, whereas larger K yields diminishing empirical gains, consistent with increasing redundancy among additional samples. Because the source-level synthetic weight is normalized independently of K, increasing K expands the candidate pool without proportionally increasing its explicit supervision weight.

Effect of source-level supervision strength. Figure 3(b) shows that performance peaks at a moderate source-level supervision strength ($m_{\text{syn}} = 1$). Smaller values provide insufficient synthetic supervision, whereas larger values give synthetic supervision excessive influence relative to observed data.

Effect of positive-support modulation. Figure 3(c) shows that SSR remains stable across a broad range of q, with $q = 1$ achieving the highest mean performance. Larger q increases the relative influence of positive-support compatibility in the composite utility, while the final normalized weights also depend on the additive smoothing term. Unless otherwise specified, we use $q = 1$.

Qualitative results. We present a qualitative example of root-cause attribution by SSR, the raw embedding detector, and naive latent synthesis (Figure 4(a)). SSR assigns the annotated root-cause step a higher score and ranking while reducing the prominence of downstream failure-correlated steps (Figure 4(b)). In addition, Figure 4(c) shows that SSR places greater supervision weight on synthetic root-cause samples that are well supported by positive examples and provide complementary, less redundant local variation.

6 Conclusion

We presented SSR, a source-conditioned supervision refinement framework that enriches scarce root-cause supervision for failure attribution. The resulting synthetic supervision is refined through source-level control and sample-level utility-guided weighting based on positive-support compatibility and source-specific non-redundancy. SSR requires no additional generation, sampling, or trajectory replay at inference time, and consistently improves root-cause attribution across datasets, language models, and evaluation settings. We hope our work will inspire future research on synthetic supervision refinement for rare-positive failure attribution in agentic systems.

AI USE STATEMENT

We used LLMs solely for grammatical correction, language polishing, and assistance with preliminary code drafting. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final manuscript and results.

REPRODUCIBILITY STATEMENT

We provide detailed implementation and experimental settings (Appendix A), additional analyses (Appendix E), and complete theoretical proofs (Appendix I) to support reproducibility.

REFERENCES

Yarden Bakish, Amir Dudai, Roy Ganz, Oren Nuriel, Elad Ben Avraham, Mor Shpigel Nacson, and Ron Litman. Adaptive influence graphs for failure attribution in multi-agent systems. *arXiv preprint arXiv:2608.24361*, 2026.

Adi Banerjee, Anirudh Nair, and Tarik Borogovac. Where did it all go wrong? a hierarchical look into multi-agent error attribution. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025.

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. Seeing the whole elephant: A benchmark for failure attribution in llm-based multi-agent systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19888–19905, 2026.

Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE transactions on neural networks and learning systems*, 34(9): 6390–6404, 2022.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025.

Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

Yu Ge, Linna Xie, Zhong Li, Yu Pei, and Tian Zhang. Who is introducing the failure? automatically attributing failures of multi-agent systems via spectrum analysis. *arXiv preprint arXiv:2509.13782*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Sungchul Hong, Seunghwan An, and Jong-June Jeon. Improving smote via fusing conditional vae for data-adaptive noise filtering. *Applied Intelligence*, 55(12):841, 2025.

Robert Hutter and Michael Pradel. Agentstepper: Interactive debugging of software development agents. *arXiv preprint arXiv:2602.06593*, 2026.

Yeonjun In, Mehrab Tanjim, Jayakumar Subramanian, Sungchul Kim, Uttaran Bhattacharya, Wonjoong Kim, Sangwu Park, Somdeb Sarkhel, and Chanyoung Park. Rethinking failure attribution in multi-agent systems: A multi-perspective benchmark and evaluation. *arXiv preprint arXiv:2603.25001*, 2026.

Jin Jia, Zhiling Deng, Zhuangbin Chen, Yingqi Wang, and Zibin Zheng. Mas-fire: Fault injection and reliability evaluation for llm-based multi-agent systems. *arXiv preprint arXiv:2602.19843*, 2026.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. Aegis: Automated error generation and attribution for multi-agent systems. *arXiv preprint arXiv:2509.14295*, 2025.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Han Li, Yifan Yao, Letian Zhu, Rili Feng, Hongyi Ye, Jiaming Wang, Yancheng He, Pengyu Zou, Lehan Zhang, Xinping Lei, et al. Codetracer: Towards traceable agent states. *arXiv preprint arXiv:2604.11641*, 2026.

Yang Liu, Hongjiang Feng, Junsong Pu, and Zhuangbin Chen. Masprism: Lightweight failure attribution for multi-agent systems using prefill-stage signals. *arXiv preprint arXiv:2605.07509*, 2026.

Guoqing Ma, Jia Zhu, Hanghui Guo, Weijie Shi, Jiawei Shen, Jingjiang Liu, and Yidan Liang. Automatic failure attribution and critical step prediction method for multi-agent systems based on causal inference. *arXiv preprint arXiv:2509.08682*, 2025.

Arnab Kumar Mondal, Lakshya Singhal, Piyush Tiwary, Parag Singla, and Prathosh AP. Minority oversampling for imbalanced data via class-preserving regularized auto-encoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 3440–3465. PMLR, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Kai Sun, Wenqiang Li, Bo Dong, Yuxin Lin, Jingyao Zhang, and Bin Shi. Scope delineation before localization: A two-stage framework for enhancing failure attribution in multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33108–33116, 2026.

Leitian Tao, Xuefeng Du, and Sharon Li. Limited preference data? learning better reward model with latent space synthesis. *Advances in Neural Information Processing Systems*, 38:165742–165772, 2025.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pp. 6438–6447. PMLR, 2019.

Alva West, Yixuan Weng, Minjun Zhu, Zhen Lin, Zhiyuan Ning, and Yue Zhang. Abduct, act, predict: Scaffolding causal inference for automated failure attribution in multi-agent systems. *arXiv preprint arXiv:2509.10401*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuxuan Yang, Hadi Akbarzadeh Khorshidi, and Uwe Aickelin. A diversity-based synthetic oversampling using clustering for handling extreme imbalance. *SN Computer Science*, 4(6):848, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yifan Yu, Moyan Li, Shaoyuan Xu, Jinmiao Fu, Xinhai Hou, Fan Lai, and Bryan Wang. Correct: Condensed error recognition via knowledge transfer in multi-agent systems. *arXiv preprint arXiv:2509.24088*, 2025.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. Agentracer: Who is inducing failure in the llm agentic systems? *arXiv preprint arXiv:2509.03312*, 2025a.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. In *International Conference on Machine Learning*, pp. 76583–76599. PMLR, 2025b.

Chenyang Zhu, Spencer Hong, Jingyu Wu, Kushal Chawla, Yuhui Tang, Youbing Yin, Nathan Wolfe, Erin Babinsky, and Daben Liu. Raffles: Reasoning-based attribution of faults for llm systems. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7659–7688, 2026a.

Taiyu Zhu, Yifan Wu, Weilin Jin, Ying Li, and Gang Huang. Stepfinder: A temporal semantic framework for failure attribution in multi-agent systems. *arXiv preprint arXiv:2606.03467*, 2026b.

A Additional Implementation Details

Experimental setup. For contextual step embedding extraction, the model input consists of the task query, the current step, and up to two preceding steps as local execution context. When fewer than two preceding steps are available, all available preceding steps are used. We use the hidden state of the final token as the fixed-dimensional representation of the current step. Our variational autoencoder (VAE) uses two-layer MLPs for both the encoder and decoder, with a hidden dimension of 64 and a latent dimension of 16. Under the default Qwen3-8B setting, the encoder maps 4,096-dimensional contextual step embeddings into the latent space, while the decoder reconstructs them back to the original embedding space. The same VAE encoder is shared across all trajectory steps, regardless of their labels, and parameterizes a diagonal Gaussian posterior for each input embedding. We use mean squared error (MSE) as the reconstruction loss.

The KL-divergence term follows the standard VAE formulation,

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{e}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right).$$

To encourage discriminative latent representations, we define three cosine-distance terms over positive and negative latent representations:

$$\mathcal{L}_{++} = \frac{1}{N_{++}} \sum_{(i,j)\in\mathcal{P}_{++}} d_{\cos}\left(\mathbf{z}_i^+, \mathbf{z}_j^+\right),$$

$$\mathcal{L}_{--} = \frac{1}{N_{--}} \sum_{(i,j)\in\mathcal{P}_{--}} d_{\cos}\left(\mathbf{z}_i^-, \mathbf{z}_j^-\right),$$

$$\mathcal{L}_{+-} = -\frac{1}{N_{+-}} \sum_{(i,j)\in\mathcal{P}_{+-}} d_{\cos}\left(\mathbf{z}_i^+, \mathbf{z}_j^-\right),$$

where $d_{\cos}$ denotes cosine distance, $\mathcal{P}_{++}$ and $\mathcal{P}_{--}$ denote the corresponding within-class pair sets, $\mathcal{P}_{+-}$ is constructed by pairing each positive representation with its nearest negative representation under cosine distance, and $N_{ab} = |\mathcal{P}_{ab}|$. The first two terms encourage within-class compactness, while $\mathcal{L}_{+-}$ encourages separation between positive and negative latent representations. The contrastive regularizer $\mathcal{L}_{\mathrm{con}}$ combines the three terms above with weights 1.8, 2.0, and 5.0 for $\mathcal{L}_{++}$, $\mathcal{L}_{--}$, and $\mathcal{L}_{+-}$, respectively. The KL-divergence and contrastive-loss weights, β and γ, are both set to 0.05. The VAE is trained for 100 epochs using Adam with a learning rate of 10^{-3} and a batch size of 128.

For latent-space synthesis, we generate $K = 5$ synthetic samples for each root-cause source using standard Gaussian perturbations with scale $\alpha = 0.1$. For source-conditioned supervision refinement, we use neighborhood size $k_{\mathrm{nn}} = 10$, geometry temperature $\tau_g = 0.25$, positive-support modulation exponent $q = 1$, and source-level supervision strength $m_{\mathrm{syn}} = 1$, unless otherwise specified. When computing d_{ij}^+, the originating root-cause representation is retained in the positive reference set.

The attribution detector is a two-layer MLP with a 512-unit hidden layer, followed by BatchNorm, ReLU activation, and dropout of 0.3, with a logistic output head. We train the detector for up to 100 epochs using SGD with an initial learning rate of 0.05, momentum 0.9, and weight decay 10^{-2}, employing early stopping with a patience of 20 epochs.

Baselines. For LLM-as-Judge (Zhang et al., 2025b), we follow the All-at-Once attribution setting described in the original paper and evaluate it in a zero-shot manner. Specifically, we perform three independent runs for each failure trajectory, with a sampling temperature of 1.0 and a maximum generation length of 4,096 tokens. For ECHO (Banerjee et al., 2025), since there is no official code open-sourced, we re-implement it following the original paper and appendix. Specifically, we use Qwen3-8B with three analysis roles, a confidence threshold of 0.3, a maximum context length of 24,000 tokens, and a maximum generation length of 1,200 tokens. For CORRECT (Yu et al., 2025), we follow the official implementation and preserve its schema-generation and retrieval pipeline. Schemas are constructed using the training partition only. Specifically, we use Qwen2.5-7B-Instruct for schema generation and diagnosis and BGE-M3 for retrieval, with $k \in \{0, 1, 3, 5, 10\}$ selected according to validation Top-1 Acc. For A2P (West et al., 2025), we follow the official implementation provided by the authors. Specifically, we use Qwen3-8B with greedy decoding, a maximum input length of 30,000 tokens, and a maximum generation length of 4,096 tokens. For MASPrism (Liu et al., 2026), since the released repository does not provide a complete evaluation pipeline, we re-implement

it following the algorithm described in the original paper. Specifically, we use Qwen3-0.6B and set the symptom ratio to 0.20, candidate size to 5, consensus coefficient to 0.30, and last-layer ratio to 0.20. For AgenTracer (Zhang et al., 2025a), we follow the GRPO-based training pipeline described in the original paper. Specifically, we adapt Qwen3-8B with LoRA using rank 16, scaling factor 32, and dropout 0.05, and train the model for two epochs with eight rollouts per instance and a learning rate of 10^{-6}. For evaluation, we use three rollouts with a sampling temperature of 0.6. For StepFinder (Zhu et al., 2026b), we follow the reported attribution architecture, which uses 128-dimensional content representations, 32-dimensional agent representations, hidden size 64, and two attention heads. For a controlled comparison, we use Qwen3-8B contextual step embeddings truncated to 128 dimensions with zero-initialized agent features.

B Results of Using Alternative Latent Representations

To verify that the gains of SSR do not merely arise from dimensionality reduction, we compare its latent representation with three alternatives: PCA, a deterministic autoencoder, and a VAE without contrastive regularization. PCA preserves maximal variance in a linear subspace, while the deterministic autoencoder removes probabilistic latent modeling and the non-contrastive VAE removes contrastive regularization. We additionally evaluate decoding the learned latent representation back to the original embedding space.

Table 3: Comparison of alternative latent representation variants. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

Representation	Probabilistic	Contrastive	Top-1 Acc.
PCA	✗	✗	41.90 ± 3.29
Deterministic AE	✗	✗	21.90 ± 6.49
VAE without contrastive regularization	✓	✗	26.35 ± 4.29
Decoder-Back	✓	✓	44.76 ± 5.19

As shown in Table 3, none of these alternative representations matches the performance of SSR in our main experiments. These results indicate that the gains of SSR cannot be explained by dimensionality reduction, deterministic compression, probabilistic modeling alone, or reconstruction back to the embedding space.

C Results with Different Dataset Splits

We verify the performance of SSR using different dataset splits. Consistent with our main experiments, we keep 10% of the trajectories for validation while varying the proportions used for training and testing. SSR achieves similar attribution performance to the results in our main experiments across the tested splits (Table 4). Meanwhile, SSR consistently outperforms the raw embedding detector and naive latent synthesis, which further verifies the robustness of our approach to different train/test partitions.

Table 4: Results with different dataset splits, reported as train/validation/test percentages. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

Method	30/10/60	40/10/50	50/10/40	60/10/30	70/10/20	80/10/10
Raw Embedding	46.32 ± 4.60	46.03 ± 1.94	46.27 ± 4.72	49.47 ± 4.71	48.46 ± 10.74	50.77 ± 18.53
Naive Latent Synthesis	38.68 ± 7.93	47.30 ± 2.84	43.14 ± 9.40	37.37 ± 11.53	41.54 ± 12.87	43.08 ± 20.06
SSR (Ours)	**48.95 ± 4.30**	**55.56 ± 1.94**	**51.76 ± 4.72**	**55.79 ± 7.06**	**57.69 ± 10.53**	**61.54 ± 16.32**

D Compute Resources and Time

Software and hardware. We conduct all experiments using Python 3.8.15 and PyTorch 2.3.1 on NVIDIA A100 GPUs with 80 GB of memory.

Training and inference time. We report the measured training and inference cost of SSR and representative baselines in Table 5. On Who&When-AG with Qwen3-8B, training the downstream

components of SSR requires approximately 2.46 seconds after representation extraction. With cached step representations, SSR requires approximately 0.050 ms of downstream attribution time per trajectory. This measurement excludes frozen-LLM embedding extraction and therefore reflects the additional inference cost introduced by the attribution model. In contrast, LLM-as-Judge and AgenTracer require generative inference, with measured end-to-end times of 27.54 s and 70.08 s, respectively, under three-rollout evaluation. These results show that SSR introduces only modest downstream training cost and requires no additional generation or trajectory replay after step representations are obtained.

Table 5: Training and inference cost on Who&When-AG using a single NVIDIA A100 80 GB GPU.

Method	**Training Time**	**Downstream Inference Time**
Raw Embedding	5.30 s	0.034 ms
Latent Mean	1.56 s	0.051 ms
Naive Latent Synthesis	2.43 s	0.050 ms
SSR (Ours)	**2.46 s**	**0.050 ms**
StepFinder	2.02 s	0.381 ms
LLM-as-Judge	–	27.54 s
AgenTracer	–	70.08 s

E Additional Ablation Studies

Ablation on geometry temperature. We study the effect of the geometry temperature τ_g on SSR by evaluating temperature values of 0.0625, 0.125, 0.25, 0.5, and 1.0. Smaller temperatures produce sharper positive-support modulation, whereas larger temperatures yield smoother sample utilities. As shown in Table 6, performance improves from small to moderate temperature values, peaking at 0.25, after which the gains plateau or slightly decrease. This demonstrates that SSR achieves robust performance without requiring a narrowly tuned geometry temperature.

Table 6: Ablation study on geometry temperature. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

τ_g	**0.0625**	**0.125**	**0.25**	**0.5**	**1.0**
Top-1 Acc.	52.70 ± 3.05	52.70 ± 3.62	$\mathbf{55.56 \pm 1.94}$	52.06 ± 2.84	52.70 ± 3.44

Ablation on neighborhood size. We study the effect of neighborhood size on the local support estimation used by SSR by evaluating neighborhood sizes of 1, 3, 5, 10, 20, 30, and 50. Larger neighborhoods provide more reference samples for estimating positive and negative support, whereas excessively large neighborhoods may weaken locality. As shown in Table 7, performance remains stable across a wide range of neighborhood sizes, peaking at $k_{\mathrm{nn}} = 10$. This indicates that SSR achieves robust performance without requiring a narrowly tuned neighborhood size.

Table 7: Ablation study on neighborhood size. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

k_{nn}	**1**	**3**	**5**	**10**	**20**	**30**	**50**
Top-1 Acc.	53.02 ± 3.29	52.38 ± 3.17	52.70 ± 2.07	$\mathbf{55.56 \pm 1.94}$	53.33 ± 3.65	52.38 ± 2.75	53.33 ± 2.66

Ablation on perturbation scale. We study the effect of the perturbation scale α on SSR by evaluating perturbation scales of 0.01, 0.05, 0.10, 0.30, 0.50, 1.00, and 2.00. Small perturbations provide limited exploration of the source-conditioned latent neighborhood, whereas excessively large perturbations may move synthetic samples away from the local structure surrounding the root-cause source. As shown in Table 8, performance improves from small to moderate perturbation scales, peaking at $\alpha = 0.10$, after which it gradually decreases. This demonstrates that SSR performs best with a moderate perturbation scale, balancing local exploration and source-conditioned consistency.

Table 8: Ablation study on perturbation scale. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

α	**0.01**	**0.05**	**0.10**	**0.30**	**0.50**	**1.00**	**2.00**
Top-1 Acc.	52.38 ± 3.17	52.06 ± 2.84	**55.56 ± 1.94**	52.38 ± 2.75	52.70 ± 2.61	53.02 ± 3.09	52.38 ± 2.24

Ablation on latent dimension. We study the effect of the latent dimension on SSR by evaluating dimensions of 8, 16, 32, 64, 128, 256, 512, 1024, and 2048. Larger latent dimensions provide greater representational capacity, whereas excessively large latent spaces may weaken the compact local structure used for synthesis and supervision refinement. As shown in Table 9, performance peaks at a latent dimension of 16, after which increasing dimensionality provides no consistent gains. This demonstrates that SSR can capture root-cause-relevant structure in a compact latent space without requiring high-dimensional latent representations.

Table 9: Ablation study on VAE latent dimension. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

d_z	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**	**2048**
Top-1 Acc.	54.29 ± 1.33	**55.56 ± 1.94**	52.70 ± 2.61	52.38 ± 4.63	53.02 ± 3.82	53.33 ± 3.09	52.38 ± 2.97	51.43 ± 2.41	52.70 ± 1.33

How do different layers impact SSR's performance?

In Figure 5, we investigate SSR using contextual embeddings extracted from different layers. All other configurations are kept the same as our main experiments. Consistent with our main findings, SSR improves failure attribution across almost all layers compared with the raw embedding detector and naive latent synthesis, indicating that SSR does not rely on a specific layer depth for improvement.

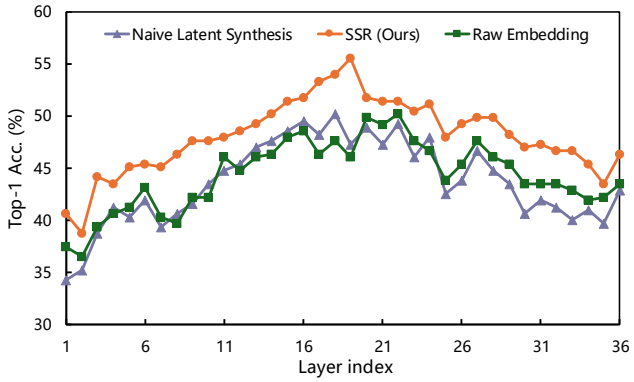


Figure 5: Failure attribution performance of SSR and controlled baselines across different layers. Results are reported on Who&When-AG using Qwen3-8B.

F Broader Model Coverage

We verify the performance of our approach on Qwen3-4B, DeepSeek-R1-Distill-Llama-8B, Qwen3-8B, and Qwen3-14B models on Who&When-AG, where SSR consistently outperforms the raw embedding detector and naive latent synthesis across all models (Table 10).

Table 10: Comparison with controlled representation and synthesis baselines across different LLM models. All values are percentages (trajectory-level Top-1 Acc.), and the best results are highlighted in **bold**.

Method	**Qwen3-4B**	**DeepSeek-R1-Distill-Llama-8B**	**Qwen3-8B**	**Qwen3-14B**
Raw Embedding	50.48 ± 3.25	40.95 ± 4.26	46.03 ± 1.94	51.43 ± 1.81
Naive Latent Synthesis	40.95 ± 6.29	37.78 ± 7.73	47.30 ± 2.84	33.97 ± 12.73
SSR (Ours)	**54.29 ± 2.61**	**49.84 ± 3.29**	**55.56 ± 1.94**	**55.87 ± 2.07**

G Additional Evaluation Metrics Beyond Top-1 Acc.

To further validate the robustness of SSR across different evaluation metrics, we additionally report AUPRC and AUROC, which are computed separately for each trajectory and then averaged across trajectories, on the main failure attribution datasets. Table 11 shows that SSR can still maintain considerable improvements over the raw embedding detector and naive latent synthesis across most evaluation settings.

Table 11: Evaluation under additional metrics beyond Top-1 Acc. across the main failure attribution datasets. We report AUPRC and AUROC.

	Who&When-AG		Who&When-HC		TraceElephant		CORRECT-Error	
Method	**AUPRC**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**AUROC**
Raw Embedding	63.31 ± 1.61	78.61 ± 1.55	21.06 ± 3.38	64.48 ± 7.50	23.85 ± 2.95	**55.64 ± 2.69**	57.84 ± 2.33	78.75 ± 2.41
Naive Latent Synthesis	62.90 ± 0.81	78.96 ± 1.47	24.28 ± 2.10	68.17 ± 6.49	20.17 ± 2.80	52.22 ± 3.13	43.18 ± 2.16	64.95 ± 1.29
SSR (Ours)	**69.46 ± 1.17**	**82.48 ± 2.06**	**34.14 ± 1.48**	**77.25 ± 5.60**	**26.06 ± 2.54**	54.59 ± 3.39	**62.54 ± 1.62**	**80.85 ± 2.79**

H Robustness to Positive-Source Scarcity

We verify the robustness of SSR under limited root-cause supervision by subsampling the available root-cause sources in the training partition while keeping the negative steps and evaluation partitions unchanged. As shown in Table 12, SSR consistently outperforms the raw embedding detector and naive latent synthesis across all supervision ratios on Who&When-AG using Qwen3-8B. These results indicate that SSR remains effective as root-cause supervision becomes increasingly scarce.

Table 12: Results under different positive-source supervision ratios. Results are reported on Who&When-AG using Qwen3-8B. All values are percentages (trajectory-level Top-1 Acc.).

Method	**25%**	**50%**	**75%**	**100%**
Raw Embedding	35.56 ± 6.41	41.27 ± 6.73	44.76 ± 3.95	46.03 ± 1.94
Naive Latent Synthesis	31.43 ± 9.08	40.00 ± 3.95	43.49 ± 6.21	47.30 ± 2.84
SSR (Ours)	**39.05 ± 8.37**	**45.71 ± 5.43**	**48.57 ± 3.98**	**55.56 ± 1.94**

I Theoretical Analysis

This appendix provides the proofs and assumptions underlying the theoretical properties stated in Section 4.2. We analyze the source-conditioned supervision refinement mechanism after the latent encoder μ_ϕ has been fixed. The analysis characterizes its source-level control property and the two structural signals used for sample-level refinement. Accordingly, the gradient analysis concerns the trainable attribution-detector parameters ω. The results establish source-level control of explicitly weighted synthetic gradients and clarify the structural interpretations of positive-support compatibility and source-specific non-redundancy. They do not claim that the proposed utility recovers a synthetic sample's true posterior probability or downstream usefulness, nor that utility-guided supervision refinement necessarily outperforms uniform allocation.

I.1 Source-Level Control of Weighted Synthetic Gradients

Recall that the synthetic samples generated from root-cause source i receive non-negative weights satisfying

$$w_{ij} \geq 0, \qquad \sum_{j=1}^{K} w_{ij} = m_{\text{syn}}. \tag{12}$$

Let $\mathbf{g}_{ij}$ denote the gradient of the BCE loss induced by $\tilde{\mathbf{z}}_{ij}$ with respect to the detector parameters under a fixed training realization. Because the latent encoder is frozen during detector training, μ_ϕ is not included in ω.

Proof of Proposition 1. By the triangle inequality, positive homogeneity of the norm, and the non-negativity of w_{ij},

$$\left\| \sum_{j=1}^{K} w_{ij} \mathbf{g}_{ij} \right\|_2 \leq \sum_{j=1}^{K} w_{ij} \left\| \mathbf{g}_{ij} \right\|_2 \tag{13}$$

$$\leq G \sum_{j=1}^{K} w_{ij} \tag{14}$$

$$= m_{\text{syn}} G, \tag{15}$$

where the second inequality uses $\left\| \mathbf{g}_{ij} \right\|_2 \leq G$, and the final equality follows from Equation (12). The bound holds for any non-negative allocation satisfying the source-level budget, regardless of synthesis multiplicity K or how the weights are distributed among synthetic samples.

Under unit weighting, $w_{ij} = 1$ for all j, and the same argument gives

$$\left\| \sum_{j=1}^{K} \mathbf{g}_{ij} \right\|_2 \leq KG. \tag{16}$$

Thus, the corresponding worst-case upper bound grows linearly with synthesis multiplicity K under unit weighting. □

Scope of the guarantee. Proposition 1 bounds the norm of the explicitly weighted synthetic-loss contributions associated with one root-cause source. It is a worst-case bound and does not imply that the realized gradient norm attains the bound, since gradients from different synthetic samples may partially cancel.

The result also does not imply identical optimization trajectories across different values of K, since candidate realizations, minibatch composition, stochastic training effects, negative-sample exposure, and the number of parameter updates may differ. Stateful components such as BatchNorm may also introduce indirect interactions that are not controlled by the source-level weight normalization.

For the minibatch-averaged detector objective, the corresponding contribution is additionally scaled by the averaging factor $1/|\mathcal{B}|$. The result therefore isolates the explicitly weighted synthetic-loss contribution rather than every consequence of changing synthesis multiplicity.

I.2 Positive-Support Compatibility

Setup. Let $d_{ij}^{+,\star}$ and $d_{ij}^{-,\star}$ denote population-level counterparts of the finite-reference positive and negative neighborhood distances used by the method. They characterize how synthetic sample $\tilde{\mathbf{z}}_{ij}$ relates to the local positive and negative regions in the underlying latent distribution. Define the corresponding population-level class-relative margin as

$$g_{ij}^{\star} = d_{ij}^{-,\star} - d_{ij}^{+,\star}. \tag{17}$$

In practice, these quantities are estimated using the mean distances d_{ij}^{+} and d_{ij}^{-} to the k_{nn} nearest positive and negative reference representations, yielding

$$g_{ij} = d_{ij}^{-} - d_{ij}^{+}. \tag{18}$$

Assumption 1 (Local reference-distance approximation)**.** *For all synthetic samples considered in the analysis, the finite-reference neighborhood distances satisfy*

$$\left| d_{ij}^{+} - d_{ij}^{+,\star} \right| \leq \delta, \qquad \left| d_{ij}^{-} - d_{ij}^{-,\star} \right| \leq \delta \tag{19}$$

for some uniform $\delta \geq 0$.

Assumption 1 abstracts the quality of the finite-reference neighborhood estimates. Its approximation error depends on factors such as local reference density, latent-space regularity, class overlap, and the choice of k_{nn}. Increasing k_{nn} may reduce sampling variance while increasing neighborhood bias; the assumption does not require the approximation error to decrease monotonically with k_{nn}.

Proposition 2 (Positive-support compatibility under bounded approximation error). *Let $g^\star_{ij} = d^{-,\star}_{ij} - d^{+,\star}_{ij}$ denote the population-level class-relative margin of synthetic sample $\tilde{\mathbf{z}}_{ij}$, and let $g_{ij} = d^-_{ij} - d^+_{ij}$ denote its finite-reference empirical counterpart used to compute the positive-support compatibility score $s_{ij} = \mathrm{sigmoid}(g_{ij}/\tau_g)$. Under Assumption 1,*

$$g_{ij} - 2\delta \;\le\; g^\star_{ij} \;\le\; g_{ij} + 2\delta. \tag{20}$$

In particular, if $g_{ij} > 2\delta$, then $g^\star_{ij} > 0$, i.e., $\tilde{\mathbf{z}}_{ij}$ is closer, under the population-level neighborhood functional, to the positive latent region than to the negative latent region. Moreover, since $\mathrm{sigmoid}(\cdot)$ is strictly increasing, s_{ij} preserves the ordering induced by the empirical margin g_{ij}. Under Assumption 1, a larger g_{ij} also yields a larger certified lower bound $g_{ij} - 2\delta$ on $g^\star_{ij}$.

Proof of Proposition 2. Using the definitions of $g^\star_{ij}$ and g_{ij},

$$\begin{aligned} \left|g^\star_{ij} - g_{ij}\right| &= \left|\left(d^{-,\star}_{ij} - d^{+,\star}_{ij}\right) - \left(d^-_{ij} - d^+_{ij}\right)\right| && (21)\\ &\le \left|d^{-,\star}_{ij} - d^-_{ij}\right| + \left|d^{+,\star}_{ij} - d^+_{ij}\right| && (22)\\ &\le 2\delta, && (23) \end{aligned}$$

where the first inequality follows from the triangle inequality and the second from Assumption 1. Therefore,

$$g_{ij} - 2\delta \le g^\star_{ij} \le g_{ij} + 2\delta. \tag{24}$$

Since the sigmoid function is strictly increasing, $s_{ij} = \mathrm{sigmoid}(g_{ij}/\tau_g)$ is monotone in both the empirical margin g_{ij} and the induced lower bound $g_{ij} - 2\delta$ on $g^\star_{ij}$. In particular, if $g_{ij} > 2\delta$, then Equation (24) gives $g^\star_{ij} > 0$, meaning that the synthetic sample is closer, under the population-level neighborhood functional, to the positive latent region than to the negative latent region. □

Scope of the result. Proposition 2 is conditional on Assumption 1. It does not establish that δ is necessarily small in the learned latent space. It also does not show that s_{ij} equals or lower-bounds the posterior probability $P(Y = 1 \mid Z = \tilde{\mathbf{z}}_{ij})$, or that synthetic samples with larger s_{ij} must yield larger downstream performance gains.

The proposition establishes a narrower statement: when finite-reference neighborhood distances approximate their population-level counterparts, g_{ij} provides a controlled surrogate for positive-support compatibility. The practical relevance of this surrogate can be evaluated by examining its relationship with independent sample-quality proxies, rather than with quantities directly reused in the construction of s_{ij}.

I.3 Properties of Source-Specific Non-Redundancy

Let $\mathcal{V}_i = \{1, \dots, K\}$ index the synthetic samples generated from source i. For any nonempty index subset $\mathcal{S} \subseteq \mathcal{V}_i$, define

$$F_i(\mathcal{S}) = \sum_{a \in \mathcal{V}_i} \max_{b \in \mathcal{S}} \mathrm{sim}\left(\tilde{\mathbf{z}}_{ia}, \tilde{\mathbf{z}}_{ib}\right). \tag{25}$$

For $K \ge 2$, the leave-one-index-out contribution is

$$\Delta^{\mathrm{nr}}_{ij} = F_i(\mathcal{V}_i) - F_i(\mathcal{V}_i \setminus \{j\}). \tag{26}$$

We use cosine similarity, $\mathrm{sim}(u, v) = u^\top v/(\|u\|_2\|v\|_2)$. Hence, $\mathrm{sim}(u, u) = 1$ and $\mathrm{sim}(u, v) \le 1$ for all nonzero u, v.

Lemma 1 (Closed-form equivalence of source-specific non-redundancy). *For $K \ge 2$, let $\mathcal{V}_i = \{1, \dots, K\}$ index the synthetic representations generated from source i. Then the leave-one-index-out facility contribution admits the closed form*

$$\Delta^{\mathrm{nr}}_{ij} = 1 - \max_{v \in \mathcal{V}_i \setminus \{j\}} \mathrm{sim}\left(\tilde{\mathbf{z}}_{ij}, \tilde{\mathbf{z}}_{iv}\right), \tag{27}$$

which coincides exactly with the source-specific non-redundancy term defined in Eq. (8).

Proof of Lemma 1. For the full index set $\mathcal{V}_i$, each query index $a \in \mathcal{V}_i$ can select itself as its representative. Since cosine self-similarity equals one and no cosine similarity exceeds one,

$$F_i(\mathcal{V}_i) = K.$$

After removing index j, every query index $a \neq j$ retains its self-match and therefore still contributes one. Only the contribution associated with query index j changes, becoming

$$\max_{v \in \mathcal{V}_i \setminus \{j\}} \mathrm{sim}\big(\tilde{\mathbf{z}}_{ij}, \tilde{\mathbf{z}}_{iv}\big).$$

Hence,

$$F_i(\mathcal{V}_i \setminus \{j\}) = (K-1) + \max_{v \in \mathcal{V}_i \setminus \{j\}} \mathrm{sim}\big(\tilde{\mathbf{z}}_{ij}, \tilde{\mathbf{z}}_{iv}\big).$$

Subtracting the two expressions yields

$$\Delta_{ij}^{\mathrm{nr}} = 1 - \max_{v \in \mathcal{V}_i \setminus \{j\}} \mathrm{sim}\big(\tilde{\mathbf{z}}_{ij}, \tilde{\mathbf{z}}_{iv}\big),$$

which proves the result. □

For the degenerate case $K = 1$, no within-source substitute exists; we therefore set $r_{i1} = 1$, and the source-normalized allocation reduces directly to $w_{i1} = m_{\mathrm{syn}}$.

Consequences. The equivalent form immediately implies:

1. $\Delta_{ij}^{\mathrm{nr}} \geq 0$;
2. if another synthetic sample in the same source-conditioned set has cosine similarity one with $\tilde{\mathbf{z}}_{ij}$, then $\Delta_{ij}^{\mathrm{nr}} = 0$;
3. $\Delta_{ij}^{\mathrm{nr}}$ increases as the maximum similarity to the closest same-source substitute decreases.

The proof also shows that $F_i(\mathcal{V}_i) \equiv K$. Thus, under the present facility definition and the cosine-similarity definition above, all leave-one-out variation is determined by the nearest same-source substitute. The non-redundancy contribution can therefore be computed directly from the pairwise similarity matrix within $\mathcal{V}_i$, without repeatedly evaluating the facility function over reduced subsets.

Interpretation. Under the stated conditions, $\Delta_{ij}^{\mathrm{nr}}$ is exactly a source-local nearest-neighbor dissimilarity. It measures whether synthetic sample $\tilde{\mathbf{z}}_{ij}$ lacks a close substitute in the source-conditioned synthetic set generated from the same annotated root-cause source. We therefore interpret it as a direct measure of source-specific non-redundancy and, more cautiously, as a local proxy for complementary source-conditioned variation. It is not a general set-level coverage measure.

This structural property does not establish positive-class validity. A synthetic sample may be highly isolated from its same-source neighbors while exhibiting low positive-support compatibility, motivating its combination with the positive-support compatibility score.

I.4 Interpretation of Utility-Guided Supervision Refinement

The proposed sample utility is

$$u_{ij} = r_{ij} s_{ij}^{q}, \qquad r_{ij} = \exp\left(\frac{\Delta_{ij}^{\mathrm{nr}}}{\tau_r}\right), \tag{28}$$

where $q \geq 0$ controls the strength of positive-support modulation.

Proposition 2 and Lemma 1 characterize the two utility components under distinct conditions:

- Under Assumption 1, s_{ij} is monotone in the certified lower bound $g_{ij} - 2\delta$ on the population-level class-relative margin $g_{ij}^{\star}$.
- Under the cosine-similarity definition above, r_{ij} is monotone in source-local nearest-neighbor dissimilarity and therefore in the absence of a close same-source substitute.

The multiplicative form allows positive-support compatibility to modulate the emphasis induced by source-specific non-redundancy. For $q > 0$, holding non-redundancy fixed, u_{ij} increases monotonically with positive-support compatibility s_{ij}; holding compatibility fixed, u_{ij} increases monotonically with source-specific non-redundancy $\Delta_{ij}^{\mathrm{nr}}$. Because the final normalization uses $u_{ij} + \varepsilon$, the exponent q modulates the unsmoothed composite utility rather than determining the final weight concentration alone.

The budget-normalized weight is

$$w_{ij} = m_{\mathrm{syn}} \frac{u_{ij} + \varepsilon}{\sum_{a=1}^{K} (u_{ia} + \varepsilon)}. \tag{29}$$

Corollary 1 (Order-preserving normalization)**.** *For any two synthetic samples j and v generated from the same root-cause source i,*

$$u_{ij} > u_{iv} \quad \Longleftrightarrow \quad w_{ij} > w_{iv}. \tag{30}$$

Proof. The normalization factor $\sum_{a=1}^{K} (u_{ia} + \varepsilon)$ in Equation (29) is strictly positive and identical for all samples in $\mathcal{C}_i$. Since w_{ij} is an increasing affine transformation of u_{ij} under this shared positive scale factor, w_{ij} and u_{ij} induce the same strict ordering among candidates generated from source i. □

Corollary 1 shows that budget normalization preserves the ordering induced by the composite utility while simultaneously enforcing $\sum_{j=1}^{K} w_{ij} = m_{\mathrm{syn}}$; the two properties are therefore not in tension, but achieved jointly by the same normalization step.

We therefore position $u_{ij} = r_{ij} s_{ij}^{q}$ as a theoretically motivated composite surrogate for positive-support compatibility and source-specific non-redundancy. The preceding results do not imply that u_{ij} estimates a synthetic sample's true posterior probability, preserves the ordering of synthetic samples' downstream marginal gains, or guarantees that utility-guided supervision refinement outperforms uniform budgeting. Such claims would require substantially stronger assumptions about the data-generating distribution, the attribution-head hypothesis class, interactions among synthetic samples, and the optimization process.

The theoretical picture is consequently limited but explicit: source-level normalization bounds the aggregate explicitly weighted synthetic-loss gradient contribution of each root-cause source independently of synthesis multiplicity (Proposition 1); positive-support compatibility provides a controlled surrogate for population-level class-relative support under a bounded local-approximation condition (Proposition 2); source-specific non-redundancy measures the absence of a close same-source substitute (Lemma 1); and utility-guided supervision refinement combines these complementary structural signals while preserving their relative ordering under source-level normalization (Corollary 1).