TIGER: Traceable Inference with Graph-Based Evidence Routing for Mitigating Hallucinations in Multimodal Generation

Anonymous ACL submission

Abstract

We study fact-level repair for multimodal generation, where a fluent output may contain specific facts that are not supported by the input. Existing inference-time repair methods often generate feedback by jointly conditioning on the input and the current output. This design has two limitations: hallucinated claims in the output can bias the model's interpretation of the input, and free-form feedback cannot be ranked or scheduled at the fact level. We present TIGER[1], an inference-time framework that redesigns feedback for localized repair. TIGER independently extracts an observation graph from the input and a claim graph from the current output, then assigns each claim a graph-conditioned risk score based on support and conflict. The model repairs selected high-risk claims while keeping the backbone frozen. We provide a convergence analysis showing that the expected total risk decreases geometrically to an explicit asymptotic bound under mild assumptions. Experiments across four cross-modal paths, including image→text, image+text→text, audio→text, and video→text, show that TIGER reduces unsupported content while preserving task quality. The gains hold across multiple backbones, and a CrisisFACTS case study suggests that the same repair mechanism can improve grounding in multi-source settings.

1 Introduction

Recent unified multimodal language models support cross-modal generation with inputs from text, images, audio, and video (Wu et al., 2024; Lu et al., 2024; Team, 2024; Zhan et al., 2024; Wu et al., 2025; Chen et al., 2025; Xu et al., 2025; Deng et al., 2025; Yang et al., 2026). This capability is useful for applications such as content creation, multimodal decision support, disaster response, medical triage, and journalism. In this paper, we focus on factual generation with textual outputs, where the input may contain one or more modalities. A key challenge in this setting is *faithfulness hallucination*, where the generated response contains facts that are not supported by the input (Zhang et al., 2024b; van Sprang et al., 2025). The response is often fluent, and most of its content may be correct, but a small number of specific facts can contradict the evidence. For example, an image with two trucks may be described as containing three trucks, an intact bridge may be described as collapsed, or "light rain" in speech may be summarized as a "heavy storm." These errors are local, but they can be harmful because downstream users may trust the generated content. Section 2 shows that such errors can arise from spurious correlations during joint autoregressive generation. *Our goal is to identify and repair unsupported facts at inference time on a frozen backbone, while preserving correct content.*

Iterative repair with feedback is a natural way to address this problem. In this setting, the model revises its current output by conditioning on the input, the current output, and a feedback signal. Existing multimodal self-correction methods, including Volcano (Lee et al., 2024), Woodpecker (Yin et al., 2024), and DeGF (Zhang et al., 2025), follow this pattern; see Appendix A. However, these methods generate feedback by jointly conditioning on the input and the current output. This design creates a failure mode: the feedback generator sees the hallucinated claims in the current output, so these claims can influence how the model interprets the input. As a result, the feedback may endorse unsupported content instead of flagging it (Fanous et al., 2025; Sun and Wang, 2026; Li et al., 2023). Free-form feedback also provides no explicit score for each claim, which makes it difficult to decide which facts should be repaired under a limited compute budget. These limitations motivate our central question:

[1]Our code is available at https://anonymous.4open.science/r/tiger-73E8

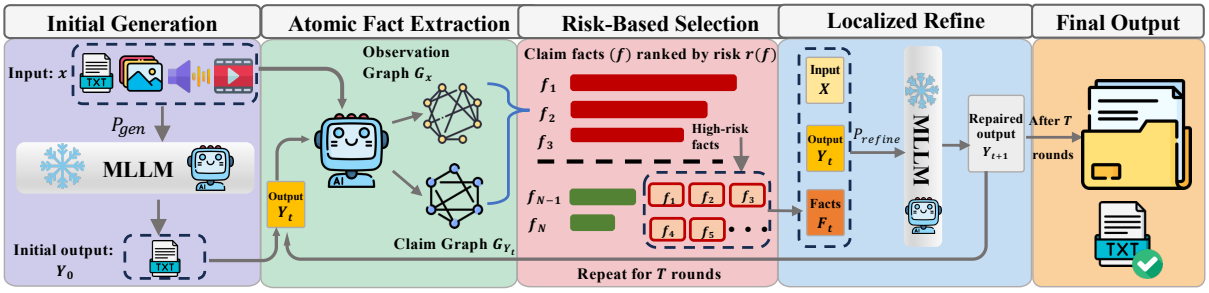


Figure 1: **Overview of TIGER.** TIGER first generates an initial output, extracts fact graphs from the input and output, ranks claims by risk, and locally repairs selected high-risk facts while keeping the backbone frozen.

Can we redesign the feedback mechanism in iterative multimodal repair so that it reduces spurious correlation during feedback generation and supports fact-level scheduling?

We answer this question with **TIGER** (**T**raceable **I**nference with **G**raph-based **E**vidence **R**outing), an inference-time framework for fact-level multimodal repair. As shown in Figure 1, TIGER redesigns feedback in two stages. First, it applies *atomic projection*: the input and the current output are extracted independently into structured fact graphs. This separation reduces the direct channel through which unsupported claims in the output can influence the model's reading of the input. Second, TIGER computes a *deterministic fact-level risk* for each output claim by measuring its support and conflict against the input graph. These scores make the feedback rankable, so TIGER can select a small set of high-risk facts and repair them locally. We also provide a convergence analysis which shows that the expected total risk decreases geometrically under mild assumptions.

We evaluate TIGER on four text-output cross-modal generation paths: image→text, image+text→text, audio→text, and video→text. The main experiments use Qwen2.5-Omni (Xu et al., 2025) as the primary frozen backbone, with additional results on LLaVA-1.5 and other open-source and proprietary backbones. Across COCO, AMBER, MMHal-Bench, Clotho, and VideoHallucer, TIGER reduces unsupported claims while preserving or improving task quality. On external hallucination metrics such as CHAIR (Rohrbach et al., 2018), TIGER mitigates the common failure mode where repair improves fluency but introduces new unsupported objects. A case study on CrisisFACTS (Buntain et al., 2023) shows that TIGER can improve grounded situation reporting from noisy evidence collected from multiple sources.

Our contributions are summarized as follows:

- We identify a feedback-stage failure mode in iterative multimodal repair: when feedback is generated by jointly conditioning on the input and the current output, hallucinated claims in the output can bias the model's interpretation of the input.
- We propose TIGER, an inference-time repair framework that replaces free-form joint feedback with independent atomic projection and deterministic fact-level risk ranking. This design makes feedback explicit, rankable, and suitable for localized repair under a fixed compute budget.
- We evaluate TIGER across four text-output cross-modal paths: image→text, image+text→text, audio→text, and video→text. Experiments show that TIGER reduces hallucination while preserving task quality, generalizes across multiple backbones, and remains effective in a real-world crisis-reporting case study.

2 A Motivating Observation

Figure 2 (top) illustrates a simple failure mode. The input image shows a person wakeboarding on water, but no boat is visible. With a free-form description prompt, Qwen2.5-Omni generates a caption that states the tow rope is attached to a boat. With an atomic enumeration prompt and the same decoding setting, the model describes the tow rope as connected to an unseen source and does not introduce the absent boat.

Figure 2 (bottom) evaluates this effect distributionally. Across four cue-to-absent object pairs, we compute the co-occurrence hallucination rate (CHR), defined as the fraction of generations that mention the absent object b when the cue a is present. We sort the pairs by their COCO co-occurrence frequency $P(b \mid a)$. Free-form generation has a higher CHR than atomic enumeration for

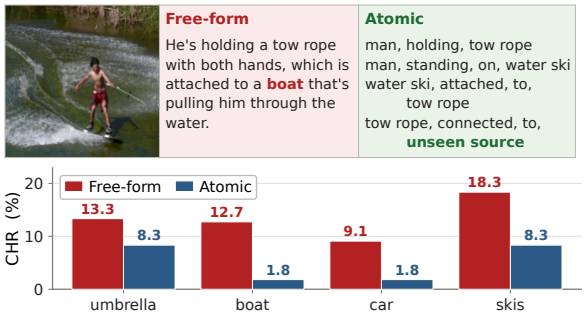


Figure 2: **Free-form generation vs. atomic enumeration.** Top: a qualitative example where free-form generation adds an unsupported object, while atomic enumeration avoids it. Bottom: co-occurrence hallucination rate (CHR) across cue-to-absent object pairs.

every pair, and atomic enumeration reduces CHR by about 2.6×. Prior work reports similar hallucination driven by co-occurrence on other backbones (Datta and Sundararaman, 2025; Li et al., 2023), which suggests that this behavior is not specific to one model. A likely mechanism is that free-form generation conditions each token on the generated text, so early claims can activate training priors and lead to unsupported content (Rohrbach et al., 2018; Zhou et al., 2024). Atomic enumeration constrains the model to produce short factual units, which reduces this prior-driven generation path.

This observation motivates TIGER. If spurious correlation enters through joint generation, a feedback mechanism that also relies on joint generation can inherit the same bias. The next section uses this observation to redesign the feedback loop with independent fact extraction and risk-based repair.

3 Methodology

3.1 Problem Setup

We consider a multimodal-to-text generation task where the input $\mathbf{X}$ can belong to any modality (image, text, audio, or video), lying in space $\mathcal{M}_X$, while the output $\mathbf{Y}$ is textual, lying in the text space $\mathcal{T}$. Given a task prompt $\mathcal{P}_{\text{gen}}$, a multimodal model Φ realizes the mapping $\mathcal{M}_X \to \mathcal{T}$:

$$\mathbf{Y} = \Phi(\mathcal{P}_{\text{gen}}, \mathbf{X}). \tag{1}$$

As shown in Section 2, single-shot generation injects spurious correlations into $\mathbf{Y}$ through joint autoregressive decoding. Prior work upgrades this into a feedback-driven iterative process:

$$\mathbf{Y}_0 = \Phi(\mathcal{P}_{\text{gen}}, \mathbf{X}),\ \mathbf{Y}_{t+1} = \Phi(\mathcal{P}_{\text{refine}}, \mathbf{X}, \mathbf{Y}_t, \mathcal{F}_t), \tag{2}$$

where $\mathcal{P}_{\text{gen}}$ is the task prompt for initial generation, $\mathcal{P}_{\text{refine}}$ is the repair prompt, and $\mathcal{F}_t$ indicates what in $\mathbf{Y}_t$ needs correction. Existing methods (Volcano, Woodpecker, DeGF) instantiate $\mathcal{F}_t$ as joint-conditioning feedback:

$$\mathcal{F}_t = \Phi(\mathcal{P}_{\text{fb}}, \mathbf{X}, \mathbf{Y}_t). \tag{3}$$

This inherits two limitations. **(i) Secondary injection of spurious correlation:** generating $\mathcal{F}_t$ via Eq. (3) is a joint autoregressive pass, so the same co-occurrence mechanism from Section 2 applies, causing $\mathcal{F}_t$ to endorse erroneous facts, not flagging them (we verify this in Section 4.5). **(ii) Non-quantifiable feedback:** $\mathcal{F}_t$ is produced by stochastic decoding and expressed in natural language, making fact-level operations such as $\arg\max$ or top-K scheduling impossible to enforce.

3.2 Redesigning $\mathcal{F}_t$

We fix limitations with two changes to Eq. (3). First, $\mathbf{X}$ and $\mathbf{Y}_t$ are each independently projected into a fact graph:

$$G_{\mathbf{X}} = \Phi(\mathcal{P}_{\text{ext}}, \mathbf{X}), \qquad G_{\mathbf{Y}_t} = \Phi(\mathcal{P}_{\text{ext}}, \mathbf{Y}_t). \tag{4}$$

Second, feedback is replaced by the output of a graph-conditioned selection operator Ψ_α on the two graphs:

$$\mathcal{F}_t = \Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t}). \tag{5}$$

Input isolation in Eq. (4) mitigates (i); graph-conditioned α-budgeted selection in Ψ_α addresses (ii). The refine step in Eq. (2) remains stochastic.

Atomic extraction. The prompt $\mathcal{P}_{\text{ext}}$ asks Φ to enumerate observable facts as triples (e_1, rel, e_2), where e_1, e_2 are head and tail entities and rel is drawn from a fixed relation vocabulary (Appendix B). Each fact is generated within the boundary of a single triple, encouraging Φ to operate in local context and reducing the co-occurrence channel documented in Section 2. The input graph $G_{\mathbf{X}}$ and the output graph $G_{\mathbf{Y}_t}$ are extracted in complete isolation. $G_{\mathbf{X}}$ is computed once before any $\mathbf{Y}$ exists and remains fixed, so errors in $\mathbf{Y}_t$ cannot directly affect the extraction of input facts, blocking the direct conditioning path behind (i). Facts that share the same entity are connected by coreference edges, so each graph carries both node-level and edge-level structure.

Risk-based selection Ψ_α. For each claim $f \in G_{\mathbf{Y}_t}$, Ψ_α computes a support score and a conflict score. The similarity between two facts is

computed as the mean of per-field clipped cosine similarities using a frozen sentence transformer (Reimers and Gurevych, 2019). The *local support* $s_0(f) = \max_{g \in G_{\mathbf{X}}} \mathrm{sim}(f, g)$ measures the best match in the observation graph. To compensate for extraction omissions, local support is propagated along coreference edges in $G_{\mathbf{Y}_t}$ with geometric decay:

$$s(f) = \max_{f' \in \{f\} \cup \mathcal{N}_K(f)} \gamma^{d(f,f')} \cdot s_0(f'), \quad (6)$$

where $\mathcal{N}_K(f)$ is the K-hop coreference neighborhood, $d(f, f')$ is the hop distance, and $\gamma \in (0, 1)$ is a decay factor. This propagation allows facts about the same entity to reinforce each other, reducing false positives caused by extraction omissions. The *conflict score* measures the strongest contradiction between a claim and the observation graph. The function $\mathrm{conflict}(f, g)$ is defined as the product of topic consistency (how well subjects and predicates match) and conclusion divergence (how much the objects differ), which acts as a soft gate: conflict is significant only when two facts discuss the same topic but reach different conclusions (Appendix C). The conflict score of a claim is:

$$c(f) = \max_{g \in G_{\mathbf{X}}} \mathrm{conflict}(f, g). \quad (7)$$

The risk combines lack of support and presence of conflict:

$$r(f) = (1 - s(f)) + \lambda \cdot c(f), \qquad \lambda > 0. \quad (8)$$

Since the scoring components are deterministic, a fixed pair $(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$ yields the same per-claim risk scores $r(f)$ for all $f \in G_{\mathbf{Y}_t}$. With $N = |G_{\mathbf{Y}_t}|$, Ψ_α returns the $\lceil \alpha N \rceil$ highest-risk facts:

$$\Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t}) = \operatorname*{arg\,max}_{\substack{\mathcal{S} \subseteq G_{\mathbf{Y}_t} \\ |\mathcal{S}| = \lceil \alpha N \rceil}} \sum_{f \in \mathcal{S}} r(f). \quad (9)$$

Each selected fact keeps its source position, so Φ repairs the corresponding span locally. The full design and property verification of $r(f)$ are given in Appendix C.

3.3 Algorithm and Convergence

The complete iterative repair procedure is summarized in Algorithm 1.

Let $R^{(t)} = \sum_{f \in G_{\mathbf{Y}_t}} r(f)$ denote the measured total risk. Under explicit assumptions (bounded graph size, per-fact repair progress ε, bounded side effects β, bounded extraction loss ξ; see Appendix D), the following conditional bound holds.

Algorithm 1 Iterative Fact-Level Repair

Require: input $\mathbf{X}$; backbone Φ; rounds T; budget $\alpha \in (0, 1]$; risk weight λ
Ensure: repaired output $\mathbf{Y}_T$

```
G_X ← Φ(P_ext, X)
Y_0 ← Φ(P_gen, X)
for t = 0, 1, ..., T − 1 do
    G_{Y_t} ← Φ(P_ext, Y_t)
    for each fact f ∈ G_{Y_t} do
        s_0(f) ← max_{g∈G_X} sim(f, g)
        s(f) ← max_{f'∈{f}∪N_K(f)} γ^{d(f,f')} · s_0(f')
        c(f) ← max_{g∈G_X} conflict(f, g)
        r(f) ← (1 − s(f)) + λ · c(f)
    end for
    F_t ← top-⌈αN⌉ facts of G_{Y_t} by r(·)
    Y_{t+1} ← Φ(P_refine, X, Y_t, F_t)
end for
return Y_T
```

Theorem 3.1 (Conditional geometric risk bound)**.** *After T rounds of repair, the expected measured risk satisfies*

$$\mathbb{E}\Big[R^{(T)}\Big] \le (1 - \alpha\varepsilon)^T R^{(0)} + \frac{\beta + \xi}{\alpha\varepsilon}. \quad (10)$$

The full proof is in Appendix D. The asymptotic floor $(\beta + \xi)/(\alpha\varepsilon)$ reflects the residual risk caused by side effects and extraction loss. Notably, because each refine step accesses raw input $\mathbf{X}$ rather than only $G_{\mathbf{X}}$, the repair loop can recover facts lost during extraction, so $\mathbf{Y}_T$ can exceed the coverage of $G_{\mathbf{X}}$. We verify this empirically in Section 4.5.

4 Experimental Evaluation

We structure our evaluation around five research questions. **RQ1** asks whether TIGER reduces hallucination across cross-modal generation paths. **RQ2** tests whether each core component of TIGER is necessary, including iterative repair, atomic projection, and deterministic risk ranking. **RQ3** studies the sensitivity of TIGER to its three main hyperparameters: the number of repair rounds T, the batch budget α, and the conflict weight λ. **RQ4** examines the mechanism behind TIGER by testing whether atomic projection reduces spurious correlation during feedback generation and whether repair can recover facts that are missing from $G_{\mathbf{X}}$. Finally, **RQ5** uses a case study to assess whether TIGER generalizes from benchmark inputs to real-world settings with noisy data from multiple sources.

Table 1: External metrics for each evaluation path.

Path	Metrics
COCO	$\text{CHAIR}_s\downarrow$, $\text{CHAIR}_i\downarrow$, BERTScore↑
AMBER	Disc. Acc.↑, $\text{CHAIR}_g\downarrow$
MMHal-Bench	BERTScore↑
Clotho	RougeL↑, BLEU↑, CLAP↑, AEHR↓
VideoHallucer	HallucRate↓, Acc_b↑, Acc_h↑, Paired↑

4.1 Experimental Setup

Datasets. We evaluate TIGER on five benchmarks that cover four cross-modal generation paths and four modalities: (1) **COCO Captions** (Chen et al., 2015) for image→text generation; (2) **MMHal-Bench** (Sun et al., 2024) for image+text→text generation; (3) **AMBER** (Wang et al., 2023) for image→text hallucination evaluation; (4) **Clotho** (Drossos et al., 2020) for audio→text generation; and (5) **VideoHallucer** (Wang et al., 2024) for video→text hallucination evaluation. We report the results as mean±std over three random seeds. Additional dataset details are provided in Appendix E.1.

Baselines. We compare TIGER with Frozen and three groups of baselines using the same frozen backbone whenever applicable. Frozen uses direct decoding from the frozen backbone without post-processing. The resampling baselines include BoN+CLIP, BoN+VisualPRM (Wang et al., 2025), and BoN+CycleReward (Bahng et al., 2025), which select one output from multiple candidates. The modality-specific decoding baselines include VCD (Leng et al., 2024) for image inputs, AAD (Hsu et al., 2025) for audio inputs, and TCD (Zhang et al., 2024a) for video inputs. The iterative refinement baselines include Woodpecker (Yin et al., 2024), DeGF (Zhang et al., 2025), and Volcano (Lee et al., 2024). The iterative refinement methods follow the joint-conditioning feedback template in Eq. (3), while the resampling and decoding methods do not produce fact-level feedback. Prompt templates and implementation details are provided in Appendices B.2 and E.2.

Metrics. To avoid self-referential evaluation, every headline metric is computed independently of TIGER's evidence graph. Table 1 lists the external metrics used for each evaluation path.

Implementation details. The primary backbone Φ is Qwen2.5-Omni-7B (Xu et al., 2025). The frozen model serves as both the generator of $\mathbf{Y}_t$ and the extractor that produces $G_{\mathbf{X}}$ and $G_{\mathbf{Y}_t}$ through Eq. (4). The extraction prompt $\mathcal{P}_{\text{ext}}$ asks Φ to enumerate facts as (e_1, rel, e_2) triples. It is applied once to $\mathbf{X}$ before the repair loop and then to each $\mathbf{Y}_t$ during repair. Per-dataset hyperparameter settings, including repair rounds T, batch budget α, and conflict weight λ, are listed in Appendix E.3. We report results with LLaVA-1.5-7B (Liu et al., 2024a) to evaluate cross-backbone generalization. Additional results with GPT-5.5, Gemini 3.5 Flash, and Claude Haiku 4.5 are reported in Appendix F.1. Appendix F.2 compares TIGER against deterministic modality-specific extractors. Per-method computational cost is summarized in Appendix F.3.

4.2 Main Results (RQ1)

To answer RQ1, we evaluate whether TIGER reduces hallucination across image→text, image+text→text, audio→text, and video→text generation paths. Table 2 reports results on COCO, AMBER, and MMHal-Bench with Qwen2.5-Omni-7B and LLaVA-1.5-7B. Table 3 reports audio→text results on Clotho, and Table 4 reports video→text results on VideoHallucer.

As shown in Table 2, TIGER reduces hallucination on image-based tasks while preserving semantic quality. On Qwen2.5-Omni-7B, it reduces COCO CHAIR_s from 0.070 to 0.050 and CHAIR_i from 0.069 to 0.035, while improving BERTScore from 0.588 to 0.643. On LLaVA-1.5-7B, it lowers COCO CHAIR_s from 0.060 to 0.030 and achieves the highest BERTScore. The same trend appears on AMBER and MMHal-Bench, where TIGER improves faithfulness without degrading text quality. These results suggest that the repair strategy is not tied to a single backbone or benchmark.

Tables 3 and 4 show that the gains extend beyond image inputs. On Clotho, TIGER improves captioning quality and audio-text alignment, and it reduces AEHR from 0.803 to 0.757, which indicates fewer unsupported audio event mentions. On VideoHallucer, TIGER lowers HallucRate from 0.015 to 0.010 and improves Paired accuracy from 0.170 to 0.198. Overall, the three tables show that TIGER consistently reduces unsupported claims across visual, acoustic, and temporal inputs while maintaining task performance.

4.3 Ablation: Are All Three Components Necessary? (RQ2)

To answer RQ2, we conduct a component ablation that tests iterative repair, atomic projection, and deterministic risk ranking under the same refinement step $\mathbf{Y}_{t+1} = \Phi(\mathcal{P}_{\text{refine}}, \mathbf{X}, \mathbf{Y}_t, \mathcal{F}_t)$. We

Table 2: Main results across two paths and two backbones. The image→text path uses COCO and AMBER ; the image+text→text path uses MMHal-Bench. **Bold**: best per column within each backbone block. Blue : TIGER.

	Qwen2.5-Omni-7B						LLaVA-1.5-7B					
	COCO			AMBER		MMHal	COCO			AMBER		MMHal
Method	CHAIR$_s$↓	CHAIR$_i$↓	BERT↑	Disc. Acc↑	CHAIR$_g$↓	BERT↑	CHAIR$_s$↓	CHAIR$_i$↓	BERT↑	Disc. Acc↑	CHAIR$_g$↓	BERT↑
Frozen	$.070_{\pm.008}$	$.069_{\pm.006}$	$.588_{\pm.003}$	$77.0_{\pm1.3}$	$5.0_{\pm.3}$	$.703_{\pm.009}$	$.060_{\pm.006}$	$.028_{\pm.003}$	$.740_{\pm.003}$	$70.3_{\pm1.3}$	$8.6_{\pm.4}$	$.757_{\pm.017}$
BoN+CLIP	$.080_{\pm.005}$	$.047_{\pm.004}$	$.613_{\pm.001}$	$78.0_{\pm1.2}$	$7.4_{\pm.4}$	$.717_{\pm.007}$	$.060_{\pm.006}$	$.025_{\pm.003}$	$.746_{\pm.003}$	$73.0_{\pm1.4}$	$8.3_{\pm.3}$	$.779_{\pm.017}$
BoN+VisPRM	$.060_{\pm.008}$	$.031_{\pm.004}$	$.616_{\pm.002}$	$79.7_{\pm1.2}$	$5.5_{\pm.3}$	$.714_{\pm.021}$	$.050_{\pm.006}$	$.021_{\pm.003}$	$.746_{\pm.003}$	$74.3_{\pm1.4}$	$8.7_{\pm.4}$	$.781_{\pm.017}$
BoN+CycRew	$.130_{\pm.011}$	$.082_{\pm.007}$	$.598_{\pm.003}$	$78.0_{\pm1.2}$	$5.3_{\pm.3}$	$.719_{\pm.011}$	$.040_{\pm.003}$	$.020_{\pm.003}$	$.745_{\pm.003}$	$72.3_{\pm1.4}$	$12.2_{\pm.5}$	$.777_{\pm.016}$
Woodpecker	$.100_{\pm.009}$	$.067_{\pm.014}$	$.616_{\pm.003}$	$79.0_{\pm.8}$	$5.1_{\pm.3}$	$.706_{\pm.017}$	$.050_{\pm.007}$	$.025_{\pm.003}$	$.740_{\pm.004}$	$76.0_{\pm1.3}$	$7.9_{\pm.3}$	$.672_{\pm.025}$
DeGF	$.060_{\pm.008}$	$.033_{\pm.003}$	$.594_{\pm.003}$	$68.3_{\pm.8}$	$5.0_{\pm.3}$	$.706_{\pm.069}$	$.040_{\pm.006}$	$.018_{\pm.003}$	$.771_{\pm.003}$	$65.7_{\pm1.5}$	$8.4_{\pm.4}$	$.780_{\pm.018}$
VCD	$.060_{\pm.007}$	$.045_{\pm.004}$	$.610_{\pm.003}$	$78.5_{\pm1.2}$	$4.8_{\pm.3}$	$.720_{\pm.010}$	$.045_{\pm.006}$	$.020_{\pm.003}$	$.755_{\pm.003}$	$72.0_{\pm1.4}$	$8.0_{\pm.4}$	$.770_{\pm.018}$
Volcano	$.070_{\pm.008}$	$.042_{\pm.004}$	$.637_{\pm.004}$	$78.0_{\pm.8}$	$5.2_{\pm.3}$	$.742_{\pm.008}$	$.040_{\pm.006}$	$.023_{\pm.004}$	$.753_{\pm.004}$	$69.3_{\pm1.5}$	$8.2_{\pm.3}$	$.767_{\pm.019}$
TIGER	$\mathbf{.050}_{\pm.007}$	$\mathbf{.035}_{\pm.005}$	$\mathbf{.643}_{\pm.003}$	$\mathbf{82.0}_{\pm.8}$	$\mathbf{4.3}_{\pm.2}$	$\mathbf{.756}_{\pm.020}$	$\mathbf{.030}_{\pm.006}$	$\mathbf{.013}_{\pm.003}$	$\mathbf{.772}_{\pm.003}$	$\mathbf{79.7}_{\pm1.5}$	$\mathbf{7.4}_{\pm.4}$	$\mathbf{.790}_{\pm.018}$

Table 3: Audio→text on Clotho with Qwen2.5-Omni-7B. CLAP is reference-free caption-audio similarity; AEHR is the audio analogue of CHAIR$_i$. **Bold**: best per column. Blue : TIGER.

Method	RougeL↑	BLEU↑	CLAP↑	AEHR↓
Frozen	$.142_{\pm.002}$	$.013_{\pm.001}$	$.332_{\pm.004}$	$.803_{\pm.009}$
BoN+CLAP	$.144_{\pm.001}$	$.016_{\pm.001}$	$.329_{\pm.003}$	$.797_{\pm.007}$
BoN+CycRew	$.139_{\pm.002}$	$.017_{\pm.000}$	$.345_{\pm.003}$	$.795_{\pm.006}$
AAD	$.140_{\pm.002}$	$.005_{\pm.000}$	$.364_{\pm.003}$	$.777_{\pm.008}$
Volcano	$.143_{\pm.001}$	$.010_{\pm.001}$	$.343_{\pm.003}$	$.800_{\pm.005}$
TIGER	$\mathbf{.147}_{\pm.001}$	$\mathbf{.019}_{\pm.001}$	$\mathbf{.376}_{\pm.004}$	$\mathbf{.757}_{\pm.008}$

Table 4: Video→text on VideoHallucer with Qwen2.5-Omni-7B. HallucRate is the analogue of CHAIR$_g$; Paired requires both halves of a pair correct. **Bold**: best per column. Blue : TIGER.

Method	HallucRate↓	Acc$_b$ ↑	Acc$_h$ ↑	Paired↑
Frozen	$.015_{\pm.003}$	$.210_{\pm.014}$	$.495_{\pm.016}$	$.170_{\pm.012}$
BoN+VideoCLIP	$.015_{\pm.004}$	$.219_{\pm.013}$	$.501_{\pm.015}$	$.175_{\pm.011}$
BoN+CycRew	$.017_{\pm.004}$	$.227_{\pm.012}$	$.508_{\pm.017}$	$.189_{\pm.013}$
TCD	$.016_{\pm.004}$	$.220_{\pm.013}$	$.505_{\pm.016}$	$.180_{\pm.012}$
Volcano	$.015_{\pm.005}$	$.250_{\pm.015}$	$.490_{\pm.017}$	$.175_{\pm.010}$
TIGER	$\mathbf{.010}_{\pm.003}$	$\mathbf{.260}_{\pm.013}$	$\mathbf{.526}_{\pm.015}$	$\mathbf{.198}_{\pm.011}$

define four levels that differ only in how the feedback signal $\mathcal{F}_t$ is produced. **L0 (Frozen)** uses direct decoding from the backbone without repair, where $\mathbf{Y} = \Phi(\mathcal{P}_{\text{gen}}, \mathbf{X})$. **L1 (Naive feedback)** adds iterative repair with joint-conditioning feedback $\mathcal{F}_t = \Phi(\mathcal{P}_{\text{fb}}, \mathbf{X}, \mathbf{Y}_t)$ as in Eq. (3). **L2 (Text feedback)** adds atomic projection by extracting $G_{\mathbf{X}}$ and $G_{\mathbf{Y}_t}$ independently through Eq. (4), but it still uses text feedback without deterministic risk ranking. **L3 (TIGER)** adds deterministic risk ranking and sets $\mathcal{F}_t = \Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$ as in Eq. (5). Thus, L0→L1 tests iterative repair alone, L1→L2 isolates atomic projection, and L2→L3 isolates deterministic risk ranking.

Figure 3 reports the ablation results on COCO. Iterative repair alone does not improve the output:

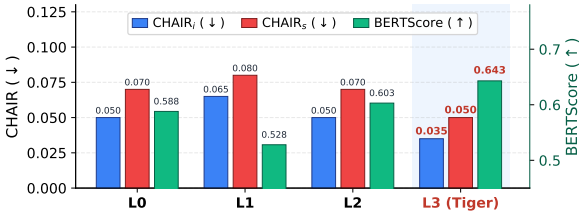


Figure 3: Component ablation on COCO.

L1 increases CHAIR$_s$ from 0.070 to 0.080 and decreases BERTScore from 0.588 to 0.528. This result supports our analysis that a repair loop can amplify hallucinations when its feedback is produced by joint conditioning on the input and output. Adding atomic projection in L2 reduces this bias and restores performance, raising BERTScore to 0.603 while bringing CHAIR$_s$ and CHAIR$_i$ back to the Frozen level. Adding deterministic risk ranking in L3 gives the best result on all three metrics, with CHAIR$_s$ = 0.050, CHAIR$_i$ = 0.035, and BERTScore = 0.643. Results show that the three components play different roles: iterative repair provides the correction mechanism, atomic projection improves the reliability of feedback, and deterministic risk ranking directs the repair budget to the facts most likely to be unsupported.

4.4 Hyperparameter Sensitivity (RQ3)

To answer RQ3, we conduct a one-at-a-time sensitivity analysis on COCO for the three main hyperparameters of TIGER: the number of repair rounds T, the repair budget α, and the conflict weight λ. For each sweep, we vary one hyperparameter and keep the other two fixed at the default setting. Figure 4 reports CHAIR$_s$, which measures object hallucination at the sentence level and is independent of the internal risk score used by TIGER.

The results show that TIGER is stable near the

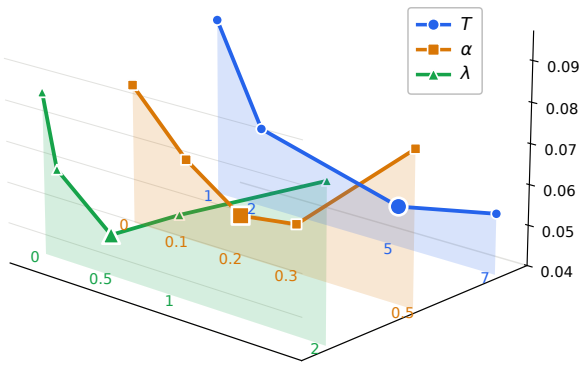


Figure 4: Hyperparameter sensitivity of TIGER on COCO. Each curve varies one hyperparameter while keeping the other two fixed. The vertical axis reports CHAIR_s; lower is better.

default setting. Increasing T reduces CHAIR_s substantially at first, but the curve flattens after a small number of repair rounds. This suggests that most correctable high-risk facts are handled early in the repair process. The repair budget α shows a trade-off between coverage and stability. A small budget may leave some risky facts unrepaired, while a large budget can revise too many facts in one round. The conflict weight λ also requires balance. When $\lambda = 0$, the selector ignores direct contradictions; when λ is too large, conflict signals can dominate the ranking. Overall, the default setting lies in a low-CHAIR_s region across all three sweeps, which indicates that the method is not sensitive to small changes in its main hyperparameters.

4.5 Mechanism: Projection and Refinement (RQ4)

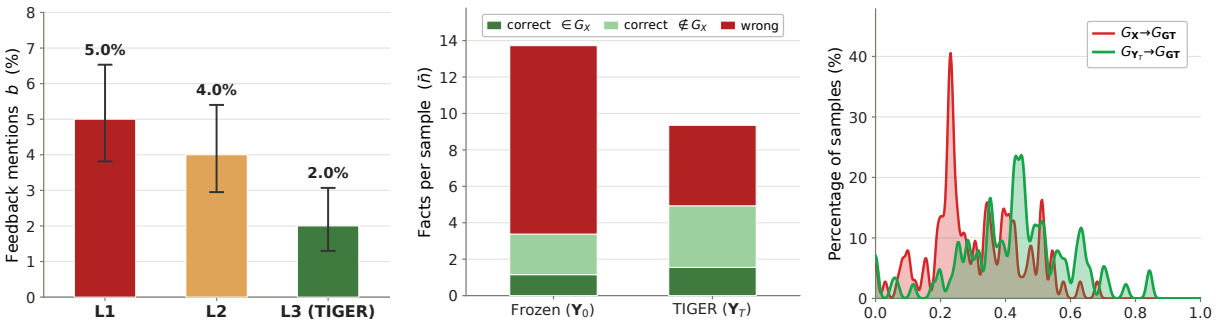
To answer RQ4, we test two mechanisms that distinguish TIGER from joint-feedback baselines: whether atomic projection reduces spurious correlation in feedback, and whether the refinement step can recover correct facts that are missing from the extracted input graph $G_{\mathbf{X}}$. These probes focus on the feedback stage because this is where unsupported claims can either be removed or reinforced.

We first test whether atomic projection reduces the spurious correlation that joint feedback can inherit from the backbone. We select 1000 images from COCO val2014 where each image contains a scene cue a (e.g., beach or kitchen) and does not contain an object b that often co-occurs with a in COCO captions. We compare three feedback channels under the same $T = 5$ repair loop. L1 uses naive joint feedback, $\mathcal{F}_t = \Phi(\mathcal{P}_{\text{fb}}, \mathbf{X}, \mathbf{Y}_t)$, as in Eq. (3). L2 uses independently extracted graphs as text in the feedback prompt, but it does not perform deterministic risk ranking. L3 uses the atomic feedback of TIGER, $\Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$. We report the feedback mention rate, defined as the fraction of samples whose feedback mentions the absent object b. Figure 5(a) shows that L1 mentions b in 5.0% of samples and L2 in 4.0%, while L3 mentions b in only 2.0%. This result indicates that joint feedback can reintroduce co-occurrence priors, whereas atomic feedback reduces this channel.

We next test whether refinement can recover correct content that the extractor misses. We label each fact in the repaired output as correct if it is supported by a reference graph $G_{\mathbf{GT}}$ extracted from the five captions of the image, and as in-extraction if it is supported by $G_{\mathbf{X}}$. Figure 5(b) reports the resulting fact composition. Compared with Frozen, TIGER keeps more correct facts, recovers additional correct facts beyond $G_{\mathbf{X}}$, and removes a large portion of wrong facts. This shows that refinement does not simply copy the extracted graph. Instead, it can still use the raw input $\mathbf{X}$ to recover facts that the extractor did not capture. Figure 5(c) gives a sample-level view of the same effect. The similarity from $G_{\mathbf{X}}$ to $G_{\mathbf{GT}}$ is concentrated around lower values, while the similarity from $G_{\mathbf{Y}_T}$ to $G_{\mathbf{GT}}$ shifts toward higher values. This shift shows that the repaired output covers more reference-supported content than the extracted input graph alone. Overall, these probes explain why TIGER improves over the L1 and L2 ablations: atomic projection reduces biased feedback, and input-grounded refinement preserves the ability to recover missing correct facts.

4.6 Case Study: Multi-Source Crisis Reporting (RQ5)

To answer RQ5, we test whether TIGER transfers from clean benchmark inputs to a noisy real-world setting. We use *CrisisFACTS* (Buntain et al., 2023), a NIST benchmark that contains Twitter, Reddit, Facebook, and online-news streams with assessor-annotated gold facts. We select one peak event-day from each of five disaster types: Hurricane, Wildfire, Flood, Explosion, and Tornado. For each event, the backbone receives text streams and six high-priority images, and it generates a situation report. Figure 6 shows the Hurricane example, where the multi-source input is consolidated into an evidence graph $G_{\mathbf{X}}$ with 387 facts and 339 edges. This setting is substantially noisier and larger than the benchmark inputs, so it tests whether fact-level



(a) Feedback mention rate. (b) Fact composition of $G_{\mathbf{Y}}$. (c) Sample-level similarity to $G_{\mathbf{GT}}$.

Figure 5: Mechanism analysis on COCO val2014. (a) Joint feedback channels mention the absent object b more often than the atomic channel. (b) Compared with Frozen, TIGER preserves correct facts, recovers additional correct facts beyond $G_{\mathbf{X}}$, and removes wrong facts. (c) The repaired output shifts toward higher similarity to $G_{\mathbf{GT}}$, which shows that refinement recovers content that the extractor misses.

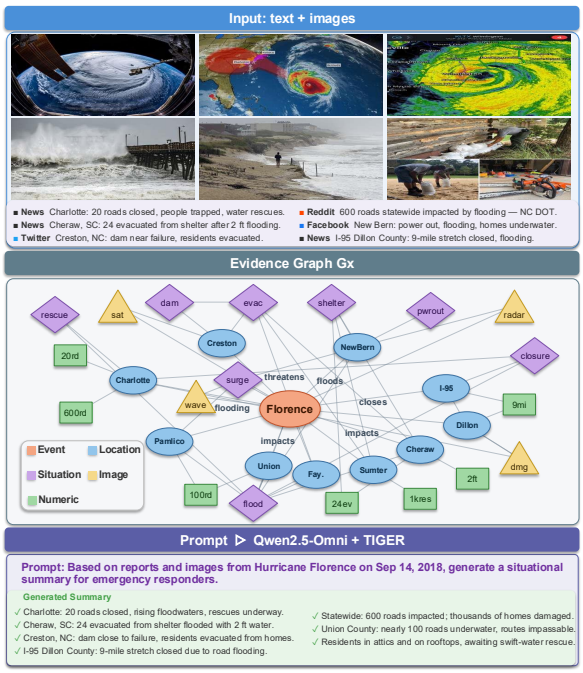


Figure 6: Case-study pipeline on the Hurricane event. Multi-source inputs are consolidated into $G_{\mathbf{X}}$.

repair remains useful beyond controlled datasets.

Table 5 reports the Frozen and TIGER results on Qwen2.5-Omni-7B. This case study tests whether the repair mechanism can improve grounding in a realistic multi-source application. Across the five disasters, TIGER improves average Precision from 0.65 to 0.96, Recall from 0.60 to 0.65, and F1 from 0.66 to 0.74. The precision gain is especially important in this setting because situation reports should avoid claims that are not supported by the evidence. At the same time, the recall and F1 improvements show that TIGER does not simply remove uncertain content; it also preserves or recovers facts that are supported by the evidence graph. Qualitatively, the repaired reports replace vague statements with grounded entities and statistics, such as changing "in some areas" to "Creston, North Carolina" and "numerous road closures" to "a 9-mile stretch of Interstate 95 in Dillon County." These results show that TIGER can support grounded summarization when the input contains noisy evidence from multiple sources.

Table 5: CrisisFACTS case study on five disaster event-days. P/R/F1 are gold-match precision, recall, and F1. **Bold**: better score. TIGER columns are shaded blue.

	P↑		R↑		F1↑	
Disaster	Frozen	TIGER	Frozen	TIGER	Frozen	TIGER
Hurricane	.679	**1.00**	.631	**.679**	.741	**.774**
Wildfire	.714	**.962**	.583	**.652**	.682	**.726**
Flood	.623	**.943**	.661	**.707**	.662	**.777**
Explosion	.583	**.909**	.552	**.594**	.588	**.687**
Tornado	.654	**.978**	.587	**.618**	.635	**.733**

5 Conclusion

We presented TIGER, an inference-time framework for repairing hallucinations in multimodal generation with a frozen backbone. TIGER independently extracts an observation graph from the input and a claim graph from the current output, then ranks claims by deterministic risk based on support and conflict. This design converts free-form feedback into a fact-level signal, enabling localized repair of high-risk claims. Experiments across four cross-modal paths show that TIGER reduces unsupported content while preserving task quality. Ablation and mechanism analyses show that iterative repair, atomic projection, and deterministic risk ranking each play a necessary role.

Limitations

This work mainly focuses on four cross-modal generation paths and evaluates TIGER on the benchmarks considered in the paper. Broader settings, such as 3D scenes, streaming multimodal inputs, and highly specialized domains, may require additional evaluation because they contain richer temporal and spatial structure than the inputs studied here. In addition, TIGER represents evidence as localized factual triples, which is well suited for objects, attributes, relations, and events, but may be less direct for abstract, causal, or subjective content. Finally, iterative repair adds inference-time computation, so future work should study more efficient scheduling strategies for real-time deployment.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

Hyojin Bahng, Caroline Chan, Fredo Durand, and Phillip Isola. 2025. Cycle consistency as reward: Learning image-text alignment without human preferences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22934–22946.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*.

Cody Buntain, Amanda Lee Hughes, Richard McCreadie, Benjamin D Horne, Muhammad Imran, and Hemant Purohit. 2023. Crisisfacts 2023-overview paper. In *TREC*.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Zuyao Chen, Jinlin Wu, Zhen Lei, and Chang Wen Chen. 2024. What makes a scene? scene graph-based evaluation and feedback for controllable generation. *arXiv preprint arXiv:2411.15435*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Shounak Datta and Dhanasekar Sundararaman. 2025. Evaluating hallucination in large vision-language models based on context-aware object similarities. *arXiv preprint arXiv:2501.15046*.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, and 1 others. 2025. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE.

Aaron Fanous, Jacob Goldberg, Ank Agarwal, Joanna Lin, Anson Zhou, Sonnet Xu, Vasiliki Bikia, Roxana Daneshjou, and Sanmi Koyejo. 2025. Syceval: Evaluating llm sycophancy. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 893–900.

Ziqi Gao, Weikai Huang, Jieyu Zhang, Aniruddha Kembhavi, and Ranjay Krishna. 2024. Generate any scene: Scene graph driven data synthesis for visual generation training. *arXiv preprint arXiv:2412.08221*.

Tzu-wen Hsu, Ke-Han Lu, Cheng-Han Chiang, and Hung-yi Lee. 2025. Reducing object hallucination in large audio-language models via audio-aware decoding. *arXiv preprint arXiv:2506.07233*.

Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. 2024. Volcano: mitigating multimodal hallucination through self-feedback guided revision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 391–404.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024b. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. 2024. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455.

Trong-Thuan Nguyen, Pha Nguyen, Jackson Cothren, Alper Yilmaz, and Khoa Luu. 2025. Hyperglm: Hypergraph for video scene graph generation and anticipation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29150–29160.

Yeji Park, Deokyeong Lee, Junsuk Choe, and Buru Chang. 2025. Convis: Contrastive decoding with hallucination visualization for mitigating hallucinations in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6434–6442.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045.

Yuan Sun and Ting Wang. 2026. Be friendly, not friends: How llm sycophancy shapes user trust. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Yuanfu Sun, Kang Li, Pengkang Guo, Jiajin Liu, and Qiaoyu Tan. 2026. Mario: Multimodal graph reasoning with large language models. *arXiv preprint arXiv:2603.05181*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, and 1 others. 2024. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13088–13110.

Junyoung Sung, Minjun Kim, Sumin An, Seungwoo Lyu, Arsha Nagrani, and Paul Hongsuck Seo. 2025. Getting to the crux: Graph-based data generation for advancing multi-hop cross-modal reasoning.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Angela van Sprang, Laurens Samson, Ana Lucic, Erman Acar, Sennay Ghebreab, and Yuki M Asano. 2025. Same content, different answers: Cross-modal inconsistency in mllms. *arXiv preprint arXiv:2512.08923*.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and 1 others. 2023. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.

Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. 2024. Videohallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and 1 others. 2025. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2024. Next-gpt: Any-to-any multimodal llm. In *Forty-first International Conference on Machine Learning*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. 2026. Mmada: Multimodal large diffusion language models. *Advances in Neural Information Processing Systems*, 38:138867–138907.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2024. Woodpecker: Hallucination correction for multimodal large language models. *Science China Information Sciences*, 67(12):220105.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, and 1 others. 2024. Anygpt:

Unified multimodal llm with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9637–9662.

Ce Zhang, Zifu Wan, Zhehan Kan, Martin Q Ma, Simon Stepputtis, Deva Ramanan, Russ Salakhutdinov, Louis-Philippe Morency, Katia Sycara, and Yaqi Xie. 2025. Self-correcting decoding with generative feedback for mitigating hallucinations in large vision-language models. *arXiv preprint arXiv:2502.06130*.

Jiacheng Zhang, Yang Jiao, Shaoxiang Chen, Na Zhao, Zhiyu Tan, Hao Li, Xingjun Ma, and Jingjing Chen. 2024a. Eventhallusion: Diagnosing event hallucinations in video llms. *arXiv preprint arXiv:2409.16597*.

Xiang Zhang, Senyu Li, Ning Shi, Bradley Hauer, Zijun Wu, Grzegorz Kondrak, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2024b. Cross-modal consistency in multimodal large language models. *arXiv preprint arXiv:2411.09273*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations*, volume 2024, pages 56969–56998.

Appendix

A Related Work **11**

B Prompts **12**
B.1 TIGER Prompts 14
B.2 Baseline Prompts 15

C Risk Function Design and Property Verification **17**
C.1 Support, Conflict, and Risk Design 17
C.2 Property Verification 18

D Convergence Proof for Iterative Risk Reduction **19**
D.1 Setup and Notation 19
D.2 Assumptions 19
D.3 Proof of Theorem 3.1 19

E Experimental Details **20**
E.1 Datasets and Preprocessing 20
E.2 Models and Baselines 20
E.3 Inference and Evaluation Protocol 20
E.4 Mechanism Analysis Methodology 21
E.5 Statistical Reporting 23
E.6 Computing Infrastructure and Software 23
E.7 Reproducibility Assets 23

F Additional Experimental Results **24**
F.1 Cross-Backbone Generalization . 24
F.2 Comparison with Deterministic Modality-Specific Extractors . . . 24
F.3 Computational Cost 24
F.4 Graph Structure and Risk-Function Ablation 24

A Related Work

Hallucination and verification. Unified multimodal models can process text, images, audio, and video, but their textual outputs may contain facts that are not supported by the input. This problem is closely related to cross-modal inconsistency, where models produce conflicting claims for the same content across modalities (Zhang et al., 2024b; van Sprang et al., 2025). Multimodal hallucination has been studied through surveys, benchmarks, and decoding methods (Bai et al., 2024; Sun et al., 2024; Wang et al., 2023; Park et al., 2025; Zhang et al., 2025). These studies provide useful evaluation tools, but many methods still score or revise the whole response, which makes it difficult to identify

the specific facts that require correction. Structured verification addresses this limitation by representing objects, attributes, and relations explicitly. Scene graph and graph based methods have been used for fine grained evaluation, controllable generation, and multimodal reasoning (Chen et al., 2024; Gao et al., 2024; Nguyen et al., 2025; Sun et al., 2026; Sung et al., 2025). TIGER follows this direction, but uses structured representations for iterative repair. It extracts an observation graph from the input and a claim graph from the current output, then compares each claim against the input through support and conflict scores.

Repair at inference time. Several methods reduce hallucination without updating the backbone. Resampling methods select one output from multiple candidates using external signals such as visual alignment, process reward models, or cycle consistency (Wang et al., 2025; Bahng et al., 2025). These methods can improve output quality, but they do not identify which facts in the response are unsupported. Feedback based repair is closer to our setting. Woodpecker (Yin et al., 2024), Volcano (Lee et al., 2024), and DeGF (Zhang et al., 2025) revise an initial output by conditioning on both the input and the current response. This design is simple, but the current response can affect how the model interprets the input, so hallucinated claims may be reinforced rather than removed. The feedback is also written in natural language, which makes it difficult to rank facts or enforce a repair budget. TIGER redesigns this step by extracting the input and output independently, assigning each claim a graph-conditioned risk score, and repairing only the selected high risk facts. This design makes feedback quantifiable and reduces the dependence of repair on joint generation.

B Prompts

This appendix lists the verbatim prompt templates used by TIGER and by the three text-feedback baselines. All prompt templates are applicable to every backbone model evaluated in this paper (both open-source and proprietary APIs) and do not depend on any specific model architecture.

Notation follows Section 3. All methods share the same generation step under a unified iterative repair framework. The generation step produces the initial output via $\mathcal{P}_{\text{gen}}$:

$$\mathbf{Y}_0 = \Phi(\mathcal{P}_{\text{gen}}, \mathbf{X}).$$

The methods differ in how the feedback signal $\mathcal{F}_t$ is produced and how the repair prompt $\mathcal{P}_{\text{refine}}^{(\cdot)}$ is instantiated. We describe each in turn.

TIGER. TIGER decomposes feedback generation into two stages: independent extraction and graph-conditioned risk computation. First, the input and the current output are each independently projected into a structured fact graph: $G_{\mathbf{X}} = \Phi(\mathcal{P}_{\text{ext}}, \mathbf{X})$, $G_{\mathbf{Y}_t} = \Phi(\mathcal{P}_{\text{ext}}, \mathbf{Y}_t)$; $G_{\mathbf{X}}$ is extracted once before the repair loop and remains fixed. The deterministic operator Ψ_α then computes per-fact risk over the two graphs and selects the high-risk set as the feedback signal: $\mathcal{F}_t = \Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$. Finally, the repair prompt $\mathcal{P}_{\text{refine}}$ produces the revised output from the raw input, the current output, and the high-risk fact set:

$$\mathbf{Y}_{t+1} = \Phi(\mathcal{P}_{\text{refine}}, \mathbf{X}, \mathbf{Y}_t, \mathcal{F}_t).$$

Per sample, TIGER calls the backbone Φ a total of $2 + 2T$ times: $\mathcal{P}_{\text{gen}}$ once, $\mathcal{P}_{\text{ext}}$ $1{+}T$ times (once for $\mathbf{X}$ and once per round for each $\mathbf{Y}_t$), and $\mathcal{P}_{\text{refine}}$ T times.

Volcano (Lee et al., 2024). We adapt Volcano's critique-then-revise protocol to the shared Qwen2.5-Omni-7B backbone, using the released critique/revise prompt style. Volcano is a multimodal LLM fine-tuned on the authors' Volcano-130k self-feedback corpus; we do *not* retrain it. Per the original paper, each round of revision is a two-step critique-then-revise procedure: the model first produces a natural-language critique of the current response conditioned on the input, then revises the response conditioned on its own critique. Formally, in round t:

$$\mathcal{F}_t = \Phi_{\text{Volcano}}(\mathcal{P}_{\text{fb}}^{\text{V}}, \mathbf{X}, \mathbf{Y}_t),$$
$$\mathbf{Y}_{t+1} = \Phi_{\text{Volcano}}(\mathcal{P}_{\text{refine}}^{\text{V}}, \mathbf{Y}_t, \mathcal{F}_t).$$

We adopt the exact critique and revise prompt templates released in the official Volcano repository (reproduced verbatim in Appendix B.2). We use the paper's default $T = 3$ critique-revise rounds. Per sample Volcano calls Φ_{Volcano} a total of $1 + 2T$ times: the initial generation once, then T critique passes and T revise passes. For modalities outside Volcano's image+text training distribution (audio, video), we fall back to a Volcano-style critique-revise pipeline on the shared frozen Qwen2.5-Omni-7B backbone with the same prompt templates — the only change is the backbone, the critique-revise procedure and prompts are unchanged.

Woodpecker (Yin et al., 2024). We faithfully reimplement Woodpecker's five-stage hallucination correction pipeline as described in the original paper. Each stage uses a dedicated prompt template adapted from the official Woodpecker codebase; the visual-validation stage uses Grounding DINO (Liu et al., 2024b) (already loaded for our deterministic-extractor comparison in Appendix F.2) as the open-vocabulary visual detector. The pipeline runs once per sample ($T = 1$); formally:

$$\begin{aligned}\mathbf{K} &= \Phi(\mathcal{P}^{\mathrm{W}}_{\mathrm{kce}},\, \mathbf{Y}_0),\\ \mathbf{Q} &= \Phi(\mathcal{P}^{\mathrm{W}}_{\mathrm{qf}},\, \mathbf{K}),\\ \mathbf{V} &= \mathrm{DINO}(\mathbf{Q},\, \mathbf{X}),\\ \mathbf{C} &= \Phi(\mathcal{P}^{\mathrm{W}}_{\mathrm{vcg}},\, \mathbf{V}),\\ \mathbf{Y}_T &= \Phi(\mathcal{P}^{\mathrm{W}}_{\mathrm{refine}},\, \mathbf{X},\, \mathbf{Y}_0,\, \mathbf{C}),\end{aligned}$$

where $\mathbf{K}$ is the list of key concepts extracted from $\mathbf{Y}_0$, $\mathbf{Q}$ is the corresponding set of verification questions, $\mathbf{V}$ is the Grounding-DINO output (boxes and labels), and $\mathbf{C}$ is the visual-claim list that serves as the explicit feedback signal $\mathcal{F}$ for the refine step. The five stages decompose as follows:

1. **Key concept extraction.** A prompt extracts the list of object-level concepts mentioned in $\mathbf{Y}_0$ that need verification. Output: a JSON list of (entity, attribute) pairs.

2. **Question formulation.** For each extracted concept, a prompt formulates a yes/no verification question of the form "Does the image contain a {concept}?" or "Is the {entity} {attribute}?".

3. **Visual knowledge validation.** For each verification question, Grounding DINO is invoked with a text query corresponding to the concept (confidence threshold 0.35, top-5 detections per query). The detected bounding boxes and labels constitute the "visual evidence" for that question.

4. **Visual claim generation.** The detected evidence is formatted into a structured claim list, e.g. "GroundingDINO finds 2 boxes labelled {label} with scores {...}". This list becomes the explicit feedback $\mathcal{F}$ for the correction stage.

5. **Hallucination correction.** The backbone Φ is called once more with the image, the original response, and the visual claim list to produce the corrected response $\mathbf{Y}_T$.

Per sample Woodpecker calls Φ four times ($\mathcal{P}_{\mathrm{gen}}$, key-concept extraction, question formulation, hallucination correction) plus one Grounding DINO call per extracted concept. The five-stage pipeline runs once per sample rather than iteratively, matching the original Woodpecker protocol. For modalities outside Grounding DINO's coverage (audio, video), Woodpecker is not applicable and is therefore omitted from the audio/video benchmarks (Tables 3, 4). Prompts for the five stages are reproduced verbatim in Appendix B.2.

DeGF (Zhang et al., 2025). DeGF (Self-Correcting Decoding with Generative Feedback) uses a text-to-image diffusion model to generate an auxiliary visual reference from the model's initial response, then performs contrastive decoding between the original image and the generated image. We follow the official protocol with two adjustments noted below. Unlike Volcano and TIGER, DeGF runs a single repair pass ($T = 1$) and consumes the feedback at the logit level rather than the prompt level:

$$\begin{aligned}\mathbf{Y}_0 &= \Phi(\mathcal{P}_{\mathrm{gen}},\, \mathbf{X}),\\ \mathbf{X}_{\mathrm{aux}} &= \mathrm{SD}(\mathbf{Y}_0),\\ \mathbf{Y}_T &= \Phi^{\mathrm{CD}}\big(\mathcal{P}_{\mathrm{gen}},\, \mathbf{X},\, \mathbf{X}_{\mathrm{aux}};\, \alpha\big),\end{aligned}$$

where $\mathrm{SD}(\cdot)$ is a Stable-Diffusion text-to-image generator and Φ^{CD} decodes greedily from contrastive logits per DeGF:

$$\tilde{s}^{(k)} = (1{+}\alpha)\, \Phi(\mathcal{P}_{\mathrm{gen}}, \mathbf{X})^{(k)} - \alpha\, \Phi(\mathcal{P}_{\mathrm{gen}}, \mathbf{X}_{\mathrm{aux}})^{(k)},$$

at every decoding step k.

The pipeline for each sample is:

1. **Initial generation.** Produce $\mathbf{Y}_0 = \Phi(\mathcal{P}_{\mathrm{gen}}, \mathbf{X})$ as usual, where $\mathbf{X}$ is the original image I_{orig}.

2. **Generative feedback via diffusion.** Take the caption-like response $\mathbf{Y}_0$ as a prompt and synthesise an auxiliary image $I_{\mathrm{gen}} = \mathrm{SD}(\mathbf{Y}_0)$ using Stable Diffusion. We use SD-Turbo (stabilityai/sd-turbo) with 1 denoising step and the default scheduler, matching the original paper's "efficient" configuration.

3. **Contrastive decoding.** Run two forward passes of the backbone with the same text

prompt: one conditioned on I_{orig} producing logits $s^{(k)}_{\text{orig}}$ at decoding step k, the other conditioned on I_{gen} producing $s^{(k)}_{\text{gen}}$. The corrected output $\mathbf{Y}_T$ is decoded greedily from the contrastive logits

$$\tilde{s}^{(k)} = (1+\alpha)\, s^{(k)}_{\text{orig}} \;-\; \alpha\, s^{(k)}_{\text{gen}}, \qquad \alpha = 0.5,$$

matching DeGF and the same α as the original paper.

Per sample DeGF runs one diffusion call plus $1 + 2L$ backbone forward passes (where L is the number of decoded tokens), so it is comparable in cost to VCD-style contrastive decoding on the same backbone. The two protocol adjustments are: (i) we use SD-Turbo (1-step) instead of the original SD 1.5 (50-step) for tractability across 1,000 images, and (ii) we use the same frozen Qwen2.5-Omni-7B / LLaVA-1.5-7B backbones as every other baseline rather than LLaVA-1.5-13B; both deviations are documented in our code release.

Key difference from TIGER. None of the three baselines performs independent fact-graph extraction. Whether through Volcano's explicit two-step process or the fused single-step process of Woodpecker and DeGF, the generation of $\mathcal{F}_t$ relies on joint autoregressive decoding—the model simultaneously observes both $\mathbf{X}$ and $\mathbf{Y}_t$ in a single call—and therefore inherits the spurious-correlation problem analyzed in Section 2.

B.1 TIGER Prompts

$\mathcal{P}_{\textbf{gen}}$**.** TIGER does not rewrite the task prompt at the first round; the raw dataset prompt is fed to Φ verbatim. The actual prompts used in our experiments are listed below. For MMHal-Bench, Clotho, and VideoHallucer, the prompt varies per instance according to the benchmark default; we show the template form.

$\mathcal{P}_{\text{gen}}$ — COCO captions

Please describe this image in detail.

$\mathcal{P}_{\text{gen}}$ — MMHal-Bench

{per-instance question}

$\mathcal{P}_{\text{gen}}$ — Clotho

{per-instance audio question}

$\mathcal{P}_{\text{gen}}$ — VideoHallucer

{per-instance video question}

$\mathcal{P}_{\textbf{ext}}$**.** A single extraction template instantiated under modality conditioning, as described in Section 3.2. The multimodal form is applied to $\mathbf{X}$ to produce $G_{\mathbf{X}}$; the text form is applied to $\mathbf{Y}_t$ to produce $G_{\mathbf{Y}_t}$. Both forms share an identical schema (field roles, standardized predicate vocabulary, and triple output format) and differ only in their few-shot exemplars and the source-domain phrasing.

$\mathcal{P}_{\text{ext}}$ — multimodal form, applied to $\mathbf{X}$

You are a multimodal fact extraction system. Extract ALL observable facts from the provided input across all modalities (image, video, audio, text). Each fact must be a triple (subject, predicate, object).
STRICT OUTPUT FORMAT — numbered list, one triple per line, parentheses required:
1. (subject, predicate, object)
2. (subject, predicate, object)
...
FIELD ROLES:
subject: head entity, written as a bare noun phrase WITHOUT attributes. GOOD: (car, is, red). BAD: (red car, is, parked).
predicate: the relation. Use the standardized forms below when applicable.
object: tail entity, attribute value, or count.
STANDARDIZED PREDICATES:
Attributes (color, size, material, shape): `is`. e.g., (car, is, red).
Counts: `count`. e.g., (lights, count, 3).
Spatial (directional; subject is the figure, object is the ground): `on / under / above / below / left of / right of / in front of / behind / inside`.
Possession or wearing or carrying: `holding / wearing / carrying / riding`.
Existence: `exists in`. e.g., (dog, exists in, image).
Actions: bare verb (`jogging / cooking / talking`).
Example — given an image of a park with a dog and a man jogging:
1. (dog, exists in, image)
2. (dog, is, golden)
3. (man, exists in, image)
4. (man, jogging on, path)
5. (man, wearing, red shirt)
6. (dog, left of, man)
7. (trees, count, 5)
8. (sky, is, blue)
Example — given an audio clip of a busy street:
1. (cars, honking, loudly)
2. (people, talking, nearby)
3. (engine, running, idle)
4. (music, playing from, shop)
Example — given text 'The president announced a new policy on Tuesday':
1. (president, announced, new policy)
2. (announcement, happened on, Tuesday)
Rules:
Extract every object, attribute, spatial relation, action, and count you can verify.
One fact per triple. Do NOT combine multiple claims into one line.

Always put the bare entity in subject; never bundle adjectives into subject.
For spatial predicates, subject is the figure positioned relative to object.
Aim for 8–20 triples. Be thorough but only include facts grounded in the input.
Output ONLY the numbered triple list. No explanation, no headers, no prose.

$\mathcal{P}_{\text{ext}}$ — text form, applied to $\mathbf{Y}_t$

You are a fact extraction system. Read the text below and decompose it into ALL factual claims as structured triples (subject, predicate, object).
STRICT OUTPUT FORMAT — numbered list, one triple per line, parentheses required:
1. (subject, predicate, object)
2. (subject, predicate, object)
...
FIELD ROLES:
subject: head entity, written as a bare noun WITHOUT adjectives. GOOD: (car, is, red). BAD: (red car, is, parked).
predicate: the relation. Prefer the standardized forms below.
object: tail entity, attribute value, or count.
STANDARDIZED PREDICATES:
Attributes: `is`. e.g., (car, is, red).
Counts: `count`. e.g., (lights, count, 3).
Existence: `exists in`. e.g., (dog, exists in, image).
Spatial (directional; subject is figure, object is ground): `on / under / above / below / left of / right of / in front of / behind / inside`.
Possession / action: `holding / wearing / carrying / riding / using`.
Free actions: bare verb (`cooking, jogging, talking, . . .`).
Example 1 — input: 'A red car is parked near a tall building on a sunny day.'
1. (car, is, red)
2. (car, parked near, building)
3. (building, is, tall)
4. (day, is, sunny)
Example 2 — input: 'The fire hydrant cap is yellow.'
1. (fire hydrant cap, is, yellow)
Example 3 — input: 'There are three traffic lights in the image.'
1. (traffic lights, exists in, image)
2. (traffic lights, count, 3)
Example 4 — input: 'A man in a white shirt is cooking in the kitchen while holding a knife.'
1. (man, exists in, image)
2. (man, wearing, white shirt)
3. (man, cooking in, kitchen)
4. (man, holding, knife)
Example 5 — input: 'The dog is on top of the car.'
1. (dog, on, car)
Rules:
Extract every claim, even from a short single sentence.
Keep the subject a bare noun; never bundle attributes into subject.
Each claim = one triple. Do NOT combine multiple facts.
For spatial predicates, subject is the figure positioned relative to object.
Output ONLY the numbered triple list. No prose.
Now extract from this text:
{<input text >}

$\mathcal{P}_{\text{refine}}$. The high-risk set $\mathcal{F}_t = \Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$ is rendered as a natural-language list and inserted into the template below. Following the same deletion-permissive editing policy used for the text-feedback baselines in §B.2, the prompt allows removal only when a flagged claim cannot be verified and no grounded replacement is available. The key distinction is that TIGER applies this policy only to the risk-ranked claims selected by Ψ_α.

$\mathcal{P}_{\text{refine}}$ — repair $\mathbf{Y}_t \to \mathbf{Y}_{t+1}$

Below is your previous response, followed by a list of claims that have been flagged as HIGH RISK under the current evidence. A claim is flagged when it has weak support, possible conflict, or cross-modal disagreement with the input.
Your goal: produce a revised response with high factual reliability while preserving all verified content. Only re-examine the flagged claims listed below. For each flagged claim:
(a) If the claim CONTRADICTS the input, correct it to match the input.
(b) If the claim is directly supported by the input, keep it.
(c) If the claim cannot be verified and no grounded replacement is available, remove it from the response.
Do not add new factual details. Do not modify unflagged claims except for minimal edits needed to keep the response coherent. A shorter response is acceptable only when unsupported flagged claims are removed. Preserve the tone and the overall answer to the task prompt.
Previous response:
"{original_text} "
High-risk claims to re-examine:
{facts_block}
Revised response:

The slot `{original_text}` is filled with $\mathbf{Y}_t$; `{facts_block}` is filled with the verbalized $\mathcal{F}_t$.

B.2 Baseline Prompts

The calling conventions of the three text-feedback baselines are described at the beginning of this appendix. Below we list the concrete prompt templates for each baseline, annotated with their symbolic roles.

All baseline numbers in Tables 2, 3, and 4 are produced by our own runs under a shared frozen-backbone protocol — they are not copied from the original papers, whose datasets, metrics, and backbones differ from ours. Within this protocol we strive for the strongest faithful reproduction available: (i) **Volcano** uses the officially released Volcano-7B checkpoint with the official critique/revise prompts; (ii) **Woodpecker** runs the full five-stage pipeline of Yin et al. (Yin et al., 2024), with Grounding DINO as the open-vocabulary detector for the visual-validation stage and the same back-

bone Φ for the LLM stages; (iii) **DeGF** uses Stable Diffusion (SD-Turbo) for generative feedback and contrastive decoding per DeGF Eq. (4), differing from the original only by the SD variant (Turbo vs. 1.5) and by the backbone choice required to match our shared protocol; (iv) **VCD**, **AAD**, and **TCD** implement the published contrastive-decoding formula $\tilde{s} = (1+\alpha)\, s_{\text{with}} - \alpha\, s_{\text{without}}$ verbatim, with modality-specific neutralization (Gaussian-noised image / silent audio / time-shuffled video frames) taken from the corresponding original recipes; (v) **BoN+CLIP**, **BoN+VisualPRM**, and **BoN+CycleReward** use the officially released reward/scoring checkpoints as the BoN reranker. Where a baseline's original pipeline depends on modules outside our shared protocol (e.g. modality-specific tools for audio or video), we note the adaptation inline; otherwise the implementation reproduces the original method on the shared backbone.

Volcano — critique prompt $\mathcal{P}^{\text{V}}_{\text{fb}}$ (verbatim from the official Volcano repository) (Lee et al., 2024)

You are given an image and a candidate response describing it. Critique the response for any visual hallucinations, factual errors, or unsupported claims. Be specific: for each problem, name the object, attribute, count, or relation that is incorrect and explain why. Do not rewrite the response; only provide the critique.
Image:
{image}
Candidate response:
{response_text}
Critique:

Volcano — revise prompt $\mathcal{P}^{\text{V}}_{\text{refine}}$ (verbatim from the official Volcano repository) (Lee et al., 2024)

You previously generated the candidate response below and received the following critique. Revise the response to address the critique while keeping all correctly described content. Output only the revised response.
Image:
{image}
Candidate response:
{response_text}
Critique:
{critique_text}
Revised response:

Woodpecker stage 1 — key concept extraction $\mathcal{P}^{\text{W}}_{\text{kce}}$ (Yin et al., 2024)

Read the following response and list the visual concepts (objects, their attributes, counts, and pairwise relations) that it asserts about the image. Output as a JSON list of items with fields "entity", "attribute", "count". Do not infer; copy only what the response explicitly claims.
Response:
{response_text}
JSON list:

Woodpecker stage 2 — question formulation $\mathcal{P}^{\text{W}}_{\text{qf}}$ (Yin et al., 2024)

For each item in the JSON list below, formulate one short yes/no verification question of the form "Does the image contain a {entity}?" or "Is the {entity} {attribute}?" or "How many {entity} are in the image?". Output as a JSON list of {"question": ..., "query": ...} where "query" is the open-vocabulary noun phrase to detect (e.g. "red car").
JSON list of concepts:
{concepts_json}
JSON list of questions:

Woodpecker stage 3 — visual validation (Grounding DINO call, no LLM prompt) (Yin et al., 2024)

For each {"query": q} in the question list, call Grounding DINO with text query q at confidence threshold 0.35. Record the top-5 detected bounding boxes and their confidence scores. Skip an item if Grounding DINO returns no detection.

Woodpecker stage 4 — visual claim generation $\mathcal{P}^{\text{W}}_{\text{vcg}}$ (Yin et al., 2024)

Given the visual detections below, write a short factual statement about what the image contains. Use the detected counts and labels literally; do not introduce concepts that were not detected. Output as plain text, one sentence per detected concept.
Detections:
{dino_detections}
Visual claims:

Woodpecker stage 5 — hallucination correction $\mathcal{P}^{\text{W}}_{\text{refine}}$ (Yin et al., 2024)

You previously generated the response below for the given image. A visual grounding tool has produced the following factual claims about what the image actually contains. Rewrite the response so that it agrees with the visual claims: correct any contradictions, drop unsupported assertions, and keep the original phrasing wherever it already matches the visual claims.
Image:
{image}
Original response:
{response_text}
Visual claims from grounding tool:
{visual_claims}
Corrected response:

DeGF — diffusion feedback (Stable Diffusion call, no LLM prompt) (Zhang et al., 2025)

Use the initial response $\mathbf{Y}_0$ as the text prompt to Stable Diffusion (SD-Turbo, 1 denoising step, default scheduler, guidance scale 1.0) and synthesise an auxiliary image I_{gen} of resolution 512×512. This auxiliary image serves as the

generative-feedback visual reference for the contrastive-decoding step.

DeGF — contrastive decoding (no LLM prompt; logit-level operation) (Zhang et al., 2025)

At each decoding step k, compute two backbone forward passes with the same text prompt: one conditioned on the original image I_{orig} giving logits $s^{(k)}_{\text{orig}}$, and one conditioned on the diffusion-generated image I_{gen} giving $s^{(k)}_{\text{gen}}$. Decode greedily from the contrastive logits

$$\tilde{s}^{(k)} = (1+\alpha)\, s^{(k)}_{\text{orig}} \;-\; \alpha\, s^{(k)}_{\text{gen}}, \qquad \alpha = 0.5,$$

matching DeGF Eq. (4). No further LLM prompt is used; the corrected response $\mathbf{Y}_T$ is the greedy decode under $\tilde{s}^{(k)}$.

C Risk Function Design and Property Verification

This appendix presents the full design of per-fact support $s(f)$, conflict $c(f)$, and risk $r(f)$ used in TIGER (§C.1), and verifies that the risk function satisfies three natural properties (§C.2).

C.1 Support, Conflict, and Risk Design

Preliminaries. Every fact in the observation graph $G_{\mathbf{X}}$ and the claim graph $G_{\mathbf{Y}_t}$ is represented as a triple $f = (s_f, p_f, o_f)$ of subject, predicate, and object text fields, following the scene-graph tuple structure of SPICE (Anderson et al., 2016). The two graphs are not loose triple sets: facts that share the same entity are connected by coreference edges. For example, (man, wearing, red shirt) and (man, holding, coffee) share the subject "man" and are therefore linked.

We encode each text field with a frozen sentence transformer Φ_{enc} (Sentence-BERT; Reimers and Gurevych, 2019; checkpoint `all-MiniLM-L6-v2`) into an L2-normalized vector. We convert the raw cosine similarity into a clipped cosine similarity

$$\kappa(t_1, t_2) = \max\{0, \Phi_{\text{enc}}(t_1)^\top \Phi_{\text{enc}}(t_2)\} \in [0, 1], \tag{11}$$

where $\|\Phi_{\text{enc}}(\cdot)\|_2 = 1$. All subsequent field similarities use $\kappa(\cdot,\cdot)$ instead of the raw cosine.

Given two facts $f = (s_f, p_f, o_f)$ and $g = (s_g, p_g, o_g)$, let $\sigma_s = \kappa(s_f, s_g)$, $\sigma_p = \kappa(p_f, p_g)$, and $\sigma_o = \kappa(o_f, o_g)$ denote the per-field bounded similarities. The similarity between f and g is defined as the equal-weight mean of the three fields:

$$\text{sim}(f, g) \;=\; \tfrac{1}{3}(\sigma_s + \sigma_p + \sigma_o). \tag{12}$$

Since each field similarity lies in $[0, 1]$, we have $\text{sim}(f, g) \in [0, 1]$. Computing similarity per field rather than encoding the concatenated triple preserves structural information and avoids the problem that whole-triple encoding assigns nearly identical similarity to "man riding horse" and "horse riding man."

Support $s(f)$. Support measures how strongly a claim is backed by evidence in the observation graph. The computation has two steps.

First, the *local support* of each claim $f \in G_{\mathbf{Y}_t}$ is the maximum similarity over all facts in the observation graph:

$$s_0(f) \;=\; \max_{g \in G_{\mathbf{X}}} \text{sim}(f, g). \tag{13}$$

The maximum is taken because a claim needs only one matching fact in $G_{\mathbf{X}}$ to be supported. Because $\text{sim}(f, g) \in [0, 1]$, we also have $s_0(f) \in [0, 1]$.

Second, local support may be too low when the extractor misses a fact. For instance, if the input image shows a man wearing a red shirt and holding coffee but the extractor only recovers (man, holding, coffee), then the claim "wearing red shirt" receives a low s_0 and would be incorrectly flagged as high risk. To compensate for extraction omissions, we propagate local support along coreference edges in the claim graph $G_{\mathbf{Y}_t}$:

$$s(f) \;=\; \max_{f' \in \{f\} \cup \mathcal{N}_K(f)} \gamma^{\,d(f, f')} \cdot s_0(f'), \tag{14}$$

where $\mathcal{N}_K(f)$ is the K-hop coreference neighborhood of f in $G_{\mathbf{Y}_t}$ (BFS expansion), $d(f, f')$ is the hop distance, and $\gamma \in (0, 1)$ is a decay factor. The geometric decay γ^d ensures that distant neighbors contribute progressively less, so that indirect support remains local to the entity neighborhood. In the example above, "wearing red shirt" is linked to "holding coffee" through the shared entity "man" and therefore receives indirect support $\gamma \cdot s_0(\text{holding coffee})$. We set $K = 3$ and $\gamma = 0.7$ throughout all experiments. When $K = 0$, the propagation is disabled and $s(f)$ reduces to $s_0(f)$. Since $s_0(f') \in [0, 1]$ and $\gamma^{d(f,f')} \in [0, 1]$, the propagated support also satisfies $s(f) \in [0, 1]$.

Graph propagation is what distinguishes TIGER from plain triple-set comparison: without graph structure every fact is evaluated independently; with propagation, facts about the same entity reinforce each other and reduce false positives caused by extraction omissions.

Conflict $c(f)$. Conflict measures the strongest contradiction between a claim and the evidence

in the observation graph. A conflict arises when two facts discuss the same topic (matching subject and predicate) but reach different conclusions (different object), corresponding to the definition of *contradiction* in natural language inference: a contradiction requires the same premise but an opposing conclusion, as opposed to *neutral* (unrelated topics).

Given f and g, conflict is defined as the product of topic consistency and conclusion divergence:

$$\text{conflict}(f, g) = \underbrace{\tfrac{1}{2}(\sigma_s + \sigma_p)}_{\text{topic consistency}} \cdot \underbrace{(1 - \sigma_o)}_{\text{conclusion divergence}}. \tag{15}$$

The multiplicative structure implements a natural soft gate. When the subjects or predicates do not match, the left factor is close to 0 and the conflict is suppressed regardless of how different the objects are, because the two facts do not discuss the same topic. When the subjects and predicates match but the objects also match, the right factor is close to 0 and there is no conflict because the conclusions agree. Only when the same entity and relation are paired with different conclusions do both factors take high values and produce a significant conflict. This soft gate introduces no threshold hyperparameters.

The conflict score of a claim f is the maximum conflict over all facts in the observation graph:

$$c(f) = \max_{g \in G_{\mathbf{X}}} \text{conflict}(f, g). \tag{16}$$

Unlike support, conflict is *not* propagated along coreference edges. Conflict is specific to an individual claim and does not transfer to neighbors: if $(\text{man}, \text{is}, \text{tall})$ conflicts with the observation $(\text{man}, \text{is}, \text{short})$, the neighboring fact $(\text{man}, \text{wearing}, \text{hat})$ is unaffected because hat and height are unrelated. Propagating conflict would let one erroneous fact contaminate all facts about the same entity.

Since $\sigma_s, \sigma_p, \sigma_o \in [0, 1]$, topic consistency $\frac{1}{2}(\sigma_s + \sigma_p) \in [0, 1]$ and conclusion divergence $(1 - \sigma_o) \in [0, 1]$. Therefore, $\text{conflict}(f, g) \in [0, 1]$, and the maximum over $G_{\mathbf{X}}$ guarantees $c(f) \in [0, 1]$.

Risk $r(f)$. Combining support and conflict, the per-fact risk is defined as

$$r(f) = (1 - s(f)) + \lambda \cdot c(f), \qquad \lambda > 0, \tag{17}$$

where $(1 - s(f))$ measures lack of support and $\lambda \cdot c(f)$ measures active conflict. Because $s(f)$ and $c(f)$ are computed by deterministic scoring operations after the graphs are fixed, $r(\cdot)$ always returns the same value for a fixed $(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$ input, satisfying the deterministic scoring requirement of the operator Ψ_α (Section 3.2).

C.2 Property Verification

We verify that the risk function $r(s, c) = (1 - s) + \lambda c$ ($\lambda > 0$, $s, c \in [0, 1]$) satisfies three natural properties.

Proposition C.1. *The risk function $r(s, c) = (1 - s) + \lambda c$ with $\lambda > 0$ satisfies the following properties:*

(P1) Non-negativity: $r(s, c) \geq 0$ *for all* $(s, c) \in [0, 1]^2$.

(P2) Boundary condition: $r(s, c) = 0$ *if and only if* $s = 1$ *and* $c = 0$.

(P3) Monotonicity: r *is strictly decreasing in* s *and strictly increasing in* c.

Proof. (P1). From $s \in [0, 1]$ we have $1 - s \geq 0$; from $c \in [0, 1]$ and $\lambda > 0$ we have $\lambda c \geq 0$. The sum of two non-negative terms is non-negative: $r(s, c) = (1 - s) + \lambda c \geq 0$.

(P2). Sufficiency: substituting $s = 1$, $c = 0$ gives $r(1, 0) = 0 + 0 = 0$. Necessity: suppose $r(s, c) = 0$, i.e., $(1 - s) + \lambda c = 0$. By (P1) both terms are non-negative; the sum of two non-negative numbers is zero if and only if both are zero. $1 - s = 0$ gives $s = 1$; $\lambda c = 0$ with $\lambda > 0$ gives $c = 0$. Hence zero risk is equivalent to full support and no conflict: a fact is risk-free only when it has a perfect match in the observation graph and contradicts no observed fact.

(P3). $\partial r / \partial s = -1 < 0$, so r is strictly decreasing in s: higher support lowers risk. $\partial r / \partial c = \lambda > 0$, so r is strictly increasing in c: higher conflict raises risk. □

Interpretation. P1 ensures that risk scores can be used for ranking and selection. P2 gives a precise semantics to zero risk: full support and no conflict is the only condition under which a fact is marked as safe. P3 ensures that risk ranking is consistent with intuition: more support lowers risk, more conflict raises it. λ is the sole hyperparameter and controls the relative penalty of contradiction versus lack of support.

D Convergence Proof for Iterative Risk Reduction

This appendix proves Theorem 3.1, establishing the geometric convergence of the expected total risk under Algorithm 1.

D.1 Setup and Notation

At round t, the claim graph $G_{\mathbf{Y}_t}$ contains $N_t := |G_{\mathbf{Y}_t}|$ facts. The total measured risk is

$$R^{(t)} = \sum_{f \in G_{\mathbf{Y}_t}} r^{(t)}(f),$$

where $r^{(t)}(f)$ is the per-fact risk defined in Eq. (17). The high-risk set $\mathcal{F}_t$ selected at line 10 of Algorithm 1 contains the $\lceil \alpha N_t \rceil$ facts with the largest $r(\cdot)$. The total risk decomposes into the selected and retained portions:

$$R^{(t)} = \underbrace{\sum_{f \in \mathcal{F}_t} r^{(t)}(f)}_{\text{to repair}} + \underbrace{\sum_{f \in G_{\mathbf{Y}_t} \setminus \mathcal{F}_t} r^{(t)}(f)}_{\text{retained}}.$$

The repair step changes the risk of facts in $\mathcal{F}_t$ from $r^{(t)}(f)$ to $\widetilde{r}^{(t+1)}(f)$ but may also affect facts outside $\mathcal{F}_t$. We define the *ideal* post-repair total risk (assuming perfect extraction) as

$$\widetilde{R}^{(t+1)} = \sum_{f \in \mathcal{F}_t} \widetilde{r}^{(t+1)}(f) + \sum_{f \in G_{\mathbf{Y}_t} \setminus \mathcal{F}_t} r^{(t)}(f) + \Delta_t^+,$$

where $\Delta_t^+ \geq 0$ is the incremental risk that repairing $\mathcal{F}_t$ introduces on the remaining facts. The gap between the *measured* risk on the actually extracted graph $G_{\mathbf{Y}_{t+1}}$ and the ideal risk is

$$\Delta_t^{\text{ext}} := R^{(t+1)} - \widetilde{R}^{(t+1)}.$$

Combining the two definitions gives the one-step decomposition:

$$R^{(t+1)} = \sum_{f \in \mathcal{F}_t} \widetilde{r}^{(t+1)}(f) + \sum_{f \in G_{\mathbf{Y}_t} \setminus \mathcal{F}_t} r^{(t)}(f) + \Delta_t^+ + \Delta_t^{\text{ext}}. \tag{18}$$

D.2 Assumptions

Assumption D.1 (Bounded graph size)**.** *There exists a constant $N_{\max}$ such that $N_t \leq N_{\max}$ for all rounds t.*

Assumption D.2 (Per-fact repair progress)**.** *There exists a constant $\varepsilon \in (0, 1]$ such that for every $f \in \mathcal{F}_t$,*

$$\mathbb{E}\Big[\widetilde{r}^{(t+1)}(f) \mid R^{(t)}\Big] \leq (1 - \varepsilon)\, r^{(t)}(f).$$

Assumption D.3 (Bounded side effects)**.** *There exists a constant $\beta \geq 0$ such that $\mathbb{E}[\Delta_t^+ \mid R^{(t)}] \leq \beta$.*

Assumption D.4 (Bounded extraction loss)**.** *There exists a constant $\xi \geq 0$ such that $\mathbb{E}[\Delta_t^{\text{ext}} \mid R^{(t)}] \leq \xi$.*

D.3 Proof of Theorem 3.1

Proof. Taking conditional expectation of the one-step decomposition (18) given $R^{(t)}$:

$$\begin{aligned}
&\mathbb{E}\Big[R^{(t+1)} \mid R^{(t)}\Big] \\
&= \sum_{f \in \mathcal{F}_t} \mathbb{E}\Big[\widetilde{r}^{(t+1)}(f) \mid R^{(t)}\Big] + \sum_{f \in G_{\mathbf{Y}_t} \setminus \mathcal{F}_t} r^{(t)}(f) \\
&\quad + \mathbb{E}\Big[\Delta_t^+ \mid R^{(t)}\Big] + \mathbb{E}\Big[\Delta_t^{\text{ext}} \mid R^{(t)}\Big] \\
&\overset{(a)}{\leq} (1 - \varepsilon) \sum_{f \in \mathcal{F}_t} r^{(t)}(f) + \sum_{f \in G_{\mathbf{Y}_t} \setminus \mathcal{F}_t} r^{(t)}(f) \\
&\quad + \beta + \xi \\
&= R^{(t)} - \varepsilon \sum_{f \in \mathcal{F}_t} r^{(t)}(f) + \beta + \xi \\
&\overset{(b)}{\leq} (1 - \alpha\varepsilon)\, R^{(t)} + \beta + \xi,
\end{aligned} \tag{19}$$

where (a) applies Assumption D.2 to each $f \in \mathcal{F}_t$ and sums, and applies Assumptions D.3 and D.4 to the side-effect and extraction terms; (b) holds because $\mathcal{F}_t$ contains the $\lceil \alpha N_t \rceil$ highest-risk facts, so their average risk is at least the overall average $R^{(t)}/N_t$, giving $\sum_{f \in \mathcal{F}_t} r^{(t)}(f) \geq \lceil \alpha N_t \rceil \cdot R^{(t)}/N_t \geq \alpha\, R^{(t)}$.

Taking unconditional expectation of (19) yields the recurrence $\mathbb{E}[R^{(t+1)}] \leq (1-\alpha\varepsilon)\, \mathbb{E}[R^{(t)}] + \beta + \xi$. Unrolling by induction, suppose the bound holds at round t; then

$$\begin{aligned}
&\mathbb{E}\Big[R^{(t+1)}\Big] \\
&\overset{(c)}{\leq} (1 - \alpha\varepsilon)\, \mathbb{E}[R^{(t)}] + \beta + \xi \\
&\overset{(d)}{\leq} (1 - \alpha\varepsilon) \left[(1 - \alpha\varepsilon)^t R^{(0)} + (\beta + \xi) \sum_{j=0}^{t-1} (1 - \alpha\varepsilon)^j \right] \\
&\quad + \beta + \xi \\
&= (1 - \alpha\varepsilon)^{t+1} R^{(0)} + (\beta + \xi) \sum_{j=0}^{t} (1 - \alpha\varepsilon)^j,
\end{aligned}$$

where (c) is the drift inequality (19) and (d) is the inductive hypothesis. The base case $t = 0$ holds with

equality. Since $\alpha\varepsilon \in (0, 1]$, $\sum_{j=0}^{T-1}(1-\alpha\varepsilon)^j \leq 1/(\alpha\varepsilon)$. Evaluating at $t = T$ gives

$$\mathbb{E}\Big[R^{(T)}\Big] \;\leq\; (1-\alpha\varepsilon)^T\, R^{(0)} \;+\; \frac{\beta+\xi}{\alpha\varepsilon}. \qquad \square$$

Asymptotic behavior. For $\alpha\varepsilon \in (0, 1)$, $(1 - \alpha\varepsilon)^T \to 0$ as $T \to \infty$, and the bound converges to the residual $(\beta + \xi)/(\alpha\varepsilon)$, which characterizes the capability boundary of the framework.

E Experimental Details

E.1 Datasets and Preprocessing

We evaluate on five benchmarks that together cover the four cross-modal generation paths reported in the main paper: image→text (COCO, AMBER), image+text→text (MMHal-Bench), audio→text (Clotho), and video→text (VideoHallucer). We also use one curated probe set (SCS-1000) for the spurious-correlation analysis in Section 2 and the feedback-mention-rate experiment in Section 4.5. Splits are the official splits as redistributed by their upstream sources; no custom random splits are introduced. SCS-1000 is a curated *probe* set, not a train / dev / test split.

No further preprocessing is applied: images are loaded at native resolution by the chosen backbone's vision tower, audio at the dataset's native sample rate. License information for each dataset follows the upstream repository.

SCS-1000 cue pairs. SCS-1000 is the curated probe set used in Section 2 (co-occurrence hallucination rate, Figure 2) and in the feedback-mention-rate analysis in Section 4.5. We select nine (a, b) pairs of COCO object categories that co-occur frequently in COCO captions, then for each pair sample roughly 110 val2014 images that contain the cue object a but *not* the absent object b (1,000 images in total). The cue is verified present and the absent object verified absent against COCO `instances_val2014.json`. Table 7 lists the nine pairs.

E.2 Models and Baselines

Primary backbone. The primary backbone Φ is Qwen2.5-Omni-7B (`Qwen/Qwen2.5-Omni-7B`) (Xu et al., 2025), which accepts text, image, audio, and video inputs and produces text and audio outputs in a single architecture. The same model serves as both the generator of $\mathbf{Y}_t$ and the extractor that produces $G_{\mathbf{X}}$ and $G_{\mathbf{Y}_t}$ via Eq. (4).

Secondary backbones (API). Used for the cross-backbone generalization experiment in Appendix F.1 only. All API calls disable temperature and rely on the endpoint default (none of these models expose a deterministic decoding flag at the time of writing).

- GPT-5.5 (Azure OpenAI, model `gpt-5.5`).
- Gemini 3.5 Flash (Google AI Studio, model `gemini-3.5-flash`) (Comanici et al., 2025).
- Claude Haiku 4.5 (Anthropic API, model `claude-haiku-4-5`).

We additionally report results with LLaVA-1.5-7B (Liu et al., 2024a) on the same image-path benchmarks as the Qwen primary backbone, also in Appendix F.1. API access windows are listed in Appendix E.7.

Baselines compared. All baselines are reimplementations on the same Qwen2.5-Omni-7B backbone (and on LLaVA-1.5-7B where applicable), called through the same dispatcher with a different `-mode` flag, so the data path, sampling hyperparameters, and evaluation protocol are matched across methods; the per-method inference budgets differ by design (e.g., best-of-N methods draw N candidates per sample, iterative methods run T repair rounds, contrastive decoders run two forward passes per generated token) and are reported separately in Table 9. Per-baseline prompt templates are listed in Appendix B.2. We compare against ten baselines grouped into three families: resampling (BoN+CLIP, BoN+VisualPRM, BoN+CycleReward), modality-specific contrastive decoding (VCD on image, AAD on audio, TCD on video), and iterative refinement (Volcano, Woodpecker, DeGF). Frozen is the no-repair lower bound.

E.3 Inference and Evaluation Protocol

Per-dataset hyperparameters. Table 10 lists the operating point of TIGER on each dataset. The operating points for T, α, and λ are listed in Table 10. The sensitivity study in Figure 4 is conducted on COCO to assess robustness around the default setting. We fix the graph-propagation parameters to $K = 3$ and $\gamma = 0.7$ across all experiments.

Chain-of-thought and answer extraction. No chain-of-thought prompting is used. Outputs

Table 6: Datasets and splits. "Full size" is the upstream benchmark size; "# used" is the sample count reported in the main paper. For COCO and SCS-1000, CHAIR-style object presence uses COCO `instances_val2014.json` augmented with a synonym table.

Dataset	Source	Split	Full size	# used	Task
COCO val2014	official val2014 captions (Chen et al., 2015)	val (subset)	40,504	1,000	img→txt (caption)
AMBER	official query sets (Wang et al., 2023)	gen./disc./attr./rel.	1,004–7,628	1,000/task	img→txt (halluc.)
MMHal-Bench	HF (Sun et al., 2024)	test	96	96	img+txt→txt
Clotho	HF (Drossos et al., 2020)	test	1,045	1,000	audio→txt (caption)
VideoHallucer	HF (Wang et al., 2024)	test	900 pairs	900	video→txt (QA)
SCS-1000	curated COCO val2014, 9 pairs (Tab. 7)	val2014	1,000	1,000	img→txt (spur. corr.)

Table 7: The nine SCS-1000 cue pairs. Each row holds the scene cue a that is present in the image and the COCO category b that is verified absent from the image but frequently co-occurs with a in COCO captions. The co-occurrence hallucination rate (CHR) in Figure 2 is the fraction of generations that mention b when conditioned on a SCS-1000 image containing a.

Cue a (present)	Absent b (frequent co-occurrent)
beach	umbrella
bed	person
dining table	cup
grass	dog
kitchen	knife
road	car
sky	airplane
snow	skis
water	boat

Table 8: Primary backbone configuration. The backbone is frozen throughout; no parameter updates occur at any stage.

Field	Value
Model	Qwen2.5-Omni-7B
HuggingFace ID	`Qwen/Qwen2.5-Omni-7B`
Size	7B (vision + audio + text)
Class	multimodal, instruct
Tokenizer	bundled (no override)
dtype	`auto` (bf16 on Ampere+)
`device_map`	`auto` (single-GPU)
`attn_impl.`	`sdpa`
Vision / audio tower	bundled, frozen
Image resolution	native, no resize
System prompt	Qwen2.5-Omni default
Max sequence length	HF default (∼32k context)

are taken verbatim from the backbone. Object-presence judgements for CHAIR are taken from the COCO `instances_val2014.json` annotations and the AMBER object-set annotations, not from the model. For MMHal-Bench we report sentence-transformer BERTScore against the reference answer rather than the original GPT-judge protocol, because BERTScore is reproducible without an external LLM call and correlates with the GPT-judge score on this benchmark. For VideoHallucer we apply a robust yes/no parser to the free-form output (regex-based, treating hedged responses such as "hard to tell" as "no"); the parser is shared across all methods in Table 4 so the comparison is fair.

Evaluation metrics. All headline metrics reported in the main paper are computed independently of TIGER's evidence graph, so the reported gains cannot be an artefact of TIGER optimizing its own internal quantities. Table 11 lists every metric used in the main paper alongside the dataset it scores and the external tool that produces it.

For the spurious-correlation probe (SCS-1000, Figure 2 in Section 2) we additionally report the Co-occurrence Hallucination Rate (CHR), defined as the fraction of generations that mention an absent object b when the cue object a is present; this follows the standard CHAIR-style fraction over the curated cue pairs.

E.4 Mechanism Analysis Methodology

This subsection documents how the three panels of Figure 5 in Section 4.5 are produced. All three probes use the SCS-1000 image set (Table 7) and the Qwen2.5-Omni-7B backbone at the COCO operating point in Table 10 ($T=5$, $\alpha=0.2$, $\lambda=0.5$, $\gamma=0.7$, $K=3$).

Panel (a): Feedback mention rate. For each image in SCS-1000 the cue object a is verified present and the absent object b is verified absent against COCO `instances_val2014.json`. We run the same $T=5$ repair loop with three different feedback channels: **L1** (naive joint feedback) instantiates $\mathcal{F}_t$ via the joint-conditioning prompt of Eq. (3), i.e., $\mathcal{F}_t = \Phi(\mathcal{P}_{\text{fb}}, \mathbf{X}, \mathbf{Y}_t)$; **L2** (text feedback) first extracts $G_{\mathbf{X}}$ and $G_{\mathbf{Y}_t}$ independently via Eq. (4) and then verbalizes both graphs into the feedback prompt as natural-language text, but

Table 9: Baselines used in the main experiments. "Paths" lists the modalities each baseline is evaluated on: image = COCO/AMBER/MMHal-Bench, audio = Clotho, video = VideoHallucer, all = every benchmark. Budget is in backbone forward passes; contrastive methods (VCD, AAD, TCD) run two passes per generated *token*.

Baseline	**Implementation**	**Budget**	**Paths**
Frozen	direct decode, no repair	1 pass/sample	all
BoN+CLIP	best-of-N rerank by CLIP	N=3, T=0.7	image
BoN+VisPRM (Wang et al., 2025)	best-of-N rerank by VisualPRM	N=3	image
BoN+CycRew (Bahng et al., 2025)	best-of-N by round-trip F1	N=3	all
BoN+CLAP	best-of-N rerank by CLAP	N=3	audio
BoN+VideoCLIP	best-of-N rerank by VideoCLIP	N=3	video
VCD (Leng et al., 2024)	contrast vs. noised image	2 pass/token	image
AAD (Hsu et al., 2025)	contrast vs. silent audio	2 pass/token	audio
TCD (Zhang et al., 2024a)	contrast vs. shuffled frames	2 pass/token	video
Volcano (Lee et al., 2024)	Volcano, critique+revise	$1+2T$ calls/sample	image (adapted A/V)
Woodpecker (Yin et al., 2024)	5-stage pipeline w/ Grounding DINO	4 calls + DINO per concept	image
DeGF (Zhang et al., 2025)	SD-Turbo aux + contrastive decoding	1 diffusion + 2 pass/token	image
TIGER (ours)	graph extraction + risk-ranked repair	$2+2T$ pass/sample	all

Table 10: Per-dataset hyperparameters of TIGER. All five benchmarks share the same Qwen2.5-Omni-7B backbone and the same sampling decoder (temperature $= 0.7$, top-$p = 0.9$), so the std reported across three seeds in Appendix E.5 reflects only the per-dataset operating-point differences. The two question-answering datasets (MMHal-Bench, VideoHallucer) use a single-fact repair regime (one flagged claim per round); the free-form datasets (COCO, AMBER, Clotho) use the batch-repair regime ($\alpha = 0.2$).

	COCO	**AMBER**	**MMHal**	**Clotho**	**VHallucer**
Rounds T	5	5	3	5	3
Budget α	0.2	0.2	single-fact	0.2	single-fact
Conflict λ	0.5	0.5	1.5	0.5	1.5
Coref γ/K	0.7 / 3	0.7 / 3	0.7 / 3	0.7 / 3	0.7 / 3
`max_new_tokens`	128	128	512	128	12
Decoding	sampling, temperature 0.7, top-p 0.9 (shared across all five benchmarks)				
Seeds	42 / 43 / 44 (results are mean±std over the three seeds)				

does not perform the deterministic risk computation Ψ_α; **L3** (TIGER) uses $\mathcal{F}_t = \Psi_\alpha(G_{\mathbf{X}}, G_{\mathbf{Y}_t})$ of Eq. (5) and verbalizes only the top-$\lceil \alpha N \rceil$ risk-ranked atomic claims. The feedback mention rate is the fraction of samples whose feedback text contains a case-insensitive whole-word match for the absent object b (plus the synonym table used by the CHAIR scorer). Error bars are 95% Wilson confidence intervals over the 1,000 binary outcomes. A lower rate means the feedback channel is less prone to re-introducing the co-occurrence prior identified in Section 2.

Panel (b): Fact composition of $G_{\mathbf{Y}}$. For each image we (i) extract $G_{\mathbf{X}}$ once from the image via Eq. (4); (ii) extract a reference graph $G_{\mathbf{GT}}$ from the five COCO ground-truth captions per image using the text form of $\mathcal{P}_{\text{ext}}$; (iii) extract $G_{\mathbf{Y}}$ from the model output (Frozen $\mathbf{Y}_0$ or TIGER $\mathbf{Y}_T$). Each fact $f \in G_{\mathbf{Y}}$ is classified into one of three disjoint bins by the sentence-transformer similarity defined in Eq. (12):

- **correct** $\in G_{\mathbf{X}}$: $\max_{g \in G_{\mathbf{GT}}} \text{sim}(f, g) \geq \tau$ *and* $\max_{g \in G_{\mathbf{X}}} \text{sim}(f, g) \geq \tau$;
- **correct** $\notin G_{\mathbf{X}}$: $\max_{g \in G_{\mathbf{GT}}} \text{sim}(f, g) \geq \tau$ *and* $\max_{g \in G_{\mathbf{X}}} \text{sim}(f, g) < \tau$;
- **wrong**: $\max_{g \in G_{\mathbf{GT}}} \text{sim}(f, g) < \tau$.

We use the same support threshold $\tau = 0.55$ as the repair loop's risk computation, so the classification rule is consistent with how the rest of the framework labels facts. The bars in panel (b) are per-sample means of the bin counts, averaged across the 1,000 SCS-1000 images. The *correct* $\notin G_{\mathbf{X}}$ slice (light green) is the key diagnostic: it counts facts that the extractor missed but the refine step recovered by reading the raw input $\mathbf{X}$ directly, which is the empirical signature of the asymptotic floor analysis in Appendix D.

Table 11: Independent (non-TIGER) evaluation metrics used in the main paper. Every metric is computed by an external tool against ground-truth annotations distributed with the benchmark, so no TIGER-internal score (support s, conflict c, risk r, or the $G_{\mathbf{X}}$ / $G_{\mathbf{Y}_t}$ graphs) enters the headline numbers.

Metric	Dataset(s)	Tool / reference
CHAIR_s, $\text{CHAIR}_i \downarrow$	COCO	COCO `instances_val2014.json` + synonym table
BERTScore $\uparrow$	COCO, MMHal-Bench	`bert_score` pkg vs. COCO captions / MMHal refs
Disc. Acc $\uparrow$	AMBER	official discriminative split, exact-match yes/no
$\text{CHAIR}_g \downarrow$	AMBER (generative)	AMBER object set per image
RougeL $\uparrow$, BLEU $\uparrow$	Clotho	`rouge_score` / `nltk` vs. 5 ref captions
CLAP $\uparrow$	Clotho	`laion/larger_clap_general` audio-text cosine
AEHR $\downarrow$	Clotho	PANNs CNN14 top-10 events vs. predicted mentions
HallucRate $\downarrow$	VideoHallucer	rate of "yes" on hallucinated questions
Acc_{b}, Acc_{h}, Paired $\uparrow$	VideoHallucer	robust yes/no parser, paired protocol

Panel (c): Sample-level similarity to $G_{\mathbf{GT}}$. For each sample i and source graph $G \in \{G_{\mathbf{X}}, G_{\mathbf{Y}_T}\}$, we compute the mean per-fact best-match similarity to $G_{\mathbf{GT}}$:

$$\bar{\sigma}_i(G) = \frac{1}{|G|} \sum_{f \in G} \max_{g \in G_{\mathbf{GT}}} \text{sim}(f, g),$$

where $\text{sim}(\cdot, \cdot)$ is the per-field cosine mean of Eq. (12). The two density curves in panel (c) are kernel density estimates of $\{\bar{\sigma}_i(G_{\mathbf{X}})\}_{i=1}^{1000}$ (red) and $\{\bar{\sigma}_i(G_{\mathbf{Y}_T})\}_{i=1}^{1000}$ (green). The bandwidth is set by Scott's rule. The rightward shift of the green curve indicates that, on average, the repaired output's claim graph covers more of the reference content than the extracted input graph alone.

E.5 Statistical Reporting

All standard deviations reported in the main paper and the appendix are computed across three decoding seeds $\{42, 43, 44\}$. We rerun each method end-to-end under each seed and report the per-method mean $\pm$ unbiased standard deviation over the three runs. The seeds are propagated via the standard `set_seed` helper to Python's `random`, NumPy, and PyTorch. Because the backbone is frozen, the only stochastic operations are the sampling decisions made during decoding of $\mathcal{P}_{\text{gen}}$, $\mathcal{P}_{\text{ext}}$, and $\mathcal{P}_{\text{refine}}$.

E.6 Computing Infrastructure and Software

Hardware. The actual GPU depends on the SLURM partition the job lands on: NVIDIA A100 , H100, or B200. Single-GPU inference is used throughout; no distributed-training framework (DDP, FSDP, DeepSpeed, Accelerate) is used because no parameter updates occur. The submission node is a 2-socket Intel Xeon Gold 6230 (80 logical cores), 752 GiB RAM, RHEL 9.6, NVIDIA driver `570.195.03`, CUDA toolkit 12.8.

Software. The main software components are listed in Table 12. FlashAttention is not installed; SDPA is used instead via `attn_implementation="sdpa"`. vLLM, DeepSpeed, PEFT, and TRL are not used. Quantization is not applied; weights are loaded in bf16 (or fp16 on non-Ampere hardware) via `dtype: auto`.

Table 12: Main software versions. Full `pip freeze` in supplementary.

Component	Version
Python / PyTorch / CUDA	3.9.21 / 2.8.0 / cu128
transformers / tokenizers	4.57.6 / 0.22.2
accelerate / huggingface_hub	1.10.1 / 0.36.2
sentence-transformers	5.1.2
torchaudio / torchvision	2.8.0 / 0.23.0
numpy / pandas	2.0.2 / 2.3.3
librosa / panns_inference	0.11.0 / 0.1.1
rouge_score / nltk	0.1.2 / 3.9.2
openai (Azure GPT-5.5)	2.30.0
google-genai (Gemini 3.5 Flash)	1.47.0
anthropic (Claude Haiku 4.5)	0.42.0

E.7 Reproducibility Assets

The complete pipeline source, evaluation configs, SLURM submission scripts, independent-metric scripts, SCS-1000 cue-pair manifest, raw per-sample outputs, and plotting scripts will be released under a permissive license at the camera-ready stage. The entry point for every experiment is

```
python -m tiger.eval -config
<yaml> -mode <mode>
```

where `<mode>` selects between TIGER, the ten baselines listed in Table 9 (Frozen, three BoN variants, three contrastive-decoding variants, three iterative-refinement variants), and the five internal ablations (`flat_baseline`, `tiger_no_graph`, `tiger_no_gy`, `tiger_no_lambda`, `tiger_no_nu`) used in Section 4.3 and Appendix F.4. Per-dataset

YAMLs under `configs/experiment/` carry the values listed in Table 10.

F Additional Experimental Results

F.1 Cross-Backbone Generalization

Table 13 reports additional COCO image→text results on three proprietary API backbones. Together with the open-source LLaVA-1.5-7B in the main results and the primary Qwen2.5-Omni-7B, this gives five backbones in total. The results show that TIGER is not tied to a specific model family. On GPT-5.5, TIGER achieves the best score on all three metrics, reducing CHAIR_s from .120 to .050 and improving BERTScore from .668 to .686. On Gemini 3.5 Flash, TIGER obtains the lowest CHAIR_s and the highest BERTScore, which suggests that the method reduces hallucination without sacrificing semantic coverage. On Claude Haiku 4.5, TIGER again achieves the lowest CHAIR_s and the highest BERTScore, while its CHAIR_i remains close to the best text-feedback variant. These trends are consistent with the main-table results on Qwen2.5-Omni-7B and LLaVA-1.5-7B.

Table 14 summarizes the Frozen-to-TIGER change across all five backbones. TIGER reduces CHAIR_s for every model, including both open-source and proprietary backbones. The relative reduction is 29% on Qwen2.5-Omni-7B, 50% on LLaVA-1.5-7B, 58% on GPT-5.5, 67% on Gemini 3.5 Flash, and 38% on Claude Haiku 4.5. BERTScore also improves on all five models. This pattern is important because CHAIR measures unsupported object mentions, while BERTScore measures semantic coverage relative to human captions. The joint improvement indicates that TIGER does not lower hallucination by deleting useful content. Instead, the independent graph extraction and deterministic risk ranking provide a model-agnostic feedback signal that transfers across both open-source and proprietary backbones.

F.2 Comparison with Deterministic Modality-Specific Extractors

TIGER uses the same frozen backbone Φ as both the generator and the fact extractor (Eq. 4). A natural alternative is to replace Φ in the extractor role with a deterministic modality-specific tool that does not autoregress, e.g., Grounding DINO for object detection, PANNs CNN14 for audio event tagging, or a generic dependency parser for text. To test whether this design choice matters, we re-ran COCO image→text with $G_{\mathbf{X}}$ produced by Grounding DINO instead of Qwen2.5-Omni-7B, keeping the rest of the pipeline (risk function, Ψ_α selection, refine step) fixed.

Grounding DINO reduces CHAIR_s to 0.055, close to the default Φ-extractor's 0.050, because it is highly reliable on the 80 COCO object classes. However, BERTScore drops from 0.643 to 0.615: a detection-only $G_{\mathbf{X}}$ contains no attributes ("red shirt"), no spatial relations ("man riding skateboard"), and no counts, so the refine step has nothing to anchor when it checks non-object claims. The same trade-off appears for the audio and video paths: PANNs covers the AudioSet ontology but discards acoustic-event ordering, and a generic dependency parser captures syntactic structure but not entity coreference across sentences. For the multimodal paths considered in this paper, using Φ itself as the extractor yields a richer $G_{\mathbf{X}}$ at the cost of a per-sample extraction call, which we judge worthwhile.

F.3 Computational Cost

Figure 7 reports wall-clock per sample on COCO val2014 for all image-path methods under the same Qwen2.5-Omni-7B backbone on a single GPU. TIGER at $T = 5$ takes $199\,\text{s}$ per sample, about $5\times$ Frozen ($38\,\text{s}$); the cost can be lowered by reducing T (the sensitivity curve in Figure 4 flattens for $T \geq 3$). Iterative methods (Volcano, Woodpecker) are in the same range as TIGER, and contrastive baselines (VCD, DeGF) are cheaper because they run only a single decoding pass.

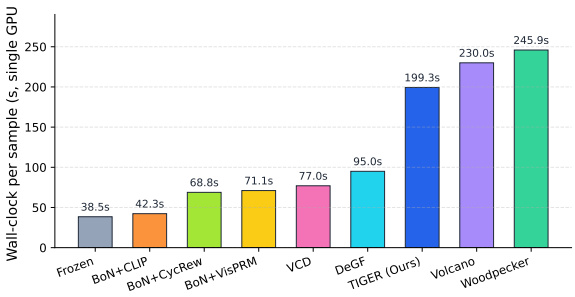


Figure 7: Wall-clock seconds per COCO sample on a single GPU (Qwen2.5-Omni-7B backbone). Bars ordered by ascending cost.

F.4 Graph Structure and Risk-Function Ablation

The L0–L3 ablation in Section 4.3 varies the feedback *paradigm* (no repair → joint feedback →

Table 13: COCO image→text results on three proprietary API backbones. **Bold**: best per column within each backbone block. Blue : TIGER.

	GPT-5.5			Gemini 3.5 Flash			Claude Haiku 4.5		
Method	$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	BERT↑	$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	BERT↑	$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	BERT↑
Frozen	.120±.010	.070±.007	.668±.003	.030±.006	.015±.003	.497±.006	.080±.009	.036±.004	.653±.003
L1 Naive fb	.120±.010	.075±.007	.672±.003	.017±.000	.010±.000	.542±.008	.105±.010	.037±.007	.660±.003
L2 Text fb	.100±.010	.065±.007	.669±.003	.015±.004	.006±.002	.459±.010	.051±.007	.027±.003	.663±.003
Woodpecker	.080±.009	.050±.006	.668±.003	.011±.003	.013±.002	.581±.007	.060±.007	.031±.004	.665±.003
DeGF	.090±.009	.058±.006	.673±.003	.020±.004	.015±.003	.535±.006	.070±.008	.031±.004	.661±.003
Volcano	.090±.009	.053±.006	.672±.003	.025±.005	.019±.004	.550±.009	.080±.009	.046±.005	.641±.003
TIGER	**.050±.007**	**.030±.004**	**.686±.003**	**.010±.003**	**.005±.002**	**.630±.009**	**.050±.007**	**.025±.004**	**.669±.003**

Table 14: Frozen-to-TIGER comparison across open-source and proprietary backbones on COCO image→text.

Backbone	Type	Frozen CHAIR_s	TIGER CHAIR_s	Δ	Frozen BERT	TIGER BERT
Qwen2.5-Omni-7B	Open	.070	**.050**	−29%	.588	**.643**
LLaVA-1.5-7B	Open	.060	**.030**	−50%	.740	**.772**
GPT-5.5	Proprietary	.120	**.050**	−58%	.668	**.686**
Gemini 3.5 Flash	Proprietary	.030	**.010**	−67%	.497	**.630**
Claude Haiku 4.5	Proprietary	.080	**.050**	−38%	.653	**.669**

Table 15: Effect of replacing the multimodal-backbone extractor with a deterministic detector (Grounding DINO) on COCO with Qwen2.5-Omni-7B. The detector reduces extraction noise on common COCO classes but discards attributes, spatial relations, and counts that the backbone-extractor captures.

Extractor for $G_{\mathbf{X}}$	$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	BERT↑
Grounding DINO (objects only)	.055	.040	.615
Qwen2.5-Omni-7B (TIGER default)	**.050**	**.035**	**.643**

Table 16: Internal-component ablation on COCO. Each row removes one component from full TIGER.

Config	$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	BERT↑
Frozen (L0)	$.070_{\pm.008}$	$.069_{\pm.006}$	$.588_{\pm.003}$
$-G_{\mathbf{Y}}$ (direct rw.)	$.065_{\pm.008}$	$.055_{\pm.006}$	$.600_{\pm.003}$
−graph (K=0)	$.063_{\pm.008}$	$.051_{\pm.006}$	$.610_{\pm.003}$
$-\lambda$ (λ=0)	$.058_{\pm.007}$	$.042_{\pm.005}$	$.628_{\pm.003}$
Full (TIGER)	$\mathbf{.050}_{\pm.007}$	$\mathbf{.035}_{\pm.005}$	$\mathbf{.643}_{\pm.003}$

atomic projection → graph-conditioned risk ranking). Here we hold the paradigm fixed at L3 (full TIGER) and instead remove one *internal* component at a time, to localize which piece of the graph machinery contributes to the final score. Table 16 reports CHAIR_s, CHAIR_i, and BERTScore on COCO val2014 under Qwen2.5-Omni-7B for three ablations of full TIGER: $-G_{\mathbf{Y}}$ **(direct rewrite)** replaces fact-level risk-driven repair with a single evidence-conditioned rewrite: the backbone receives all $G_{\mathbf{X}}$ facts and rewrites the entire output without constructing $G_{\mathbf{Y}_t}$; **−graph (flat+repair)** sets $K = 0$ in Eq. (14), so $s(f) = s_0(f)$ with no coreference propagation; $-\lambda$ (λ=0) zeros out the conflict term in the risk function so $r(f) = 1 - s(f)$. The Frozen row (the L0 baseline in Figure 3) is included for reference.

Three findings stand out. First, every ablation lies between full TIGER and Frozen on all three metrics, which confirms that the three components are jointly necessary: removing any single one gives up a fraction of the TIGER-over-Frozen gap but does not collapse the repair loop. Second, the ablations order cleanly by the component they remove. The $-\lambda$ variant (λ=0, $r(f)$=1 − $s(f)$) is closest to full TIGER; without the conflict term the selector can no longer separate weakly-supported claims from actively contradicted ones, but most flagged facts are still correctly low-support and the repair loop remains useful. The −graph variant ($K = 0$, no coreference propagation) sits slightly further from TIGER; without propagation the support score uses only direct matches, so facts that are supported only indirectly through a shared entity get over-flagged and revised away. Removing the claim graph $G_{\mathbf{Y}}$ entirely is the strongest single ablation: without $G_{\mathbf{Y}}$ the refine step has no per-claim risk to act on and falls back to a single evidence-conditioned rewrite that cannot localize edits. Third, none of the three single-component ablations matches the joint $L_0 \rightarrow L_1$ regression in Figure 3, where the full feedback paradigm is changed from atomic-with-risk to joint-conditioning. This indicates that the paradigm choice (atomic projection + risk ranking, taken as a whole) is the dominant source of TIGER's gain over self-correction baselines, while each internal component within that paradigm contributes a smaller improvement.