THE POISONED CONVERSATION: PRIVACY-LEAKING WATERMARKS IN UNIFIED MULTIMODAL MODELS

Anonymous authors
Paper under double-blind review

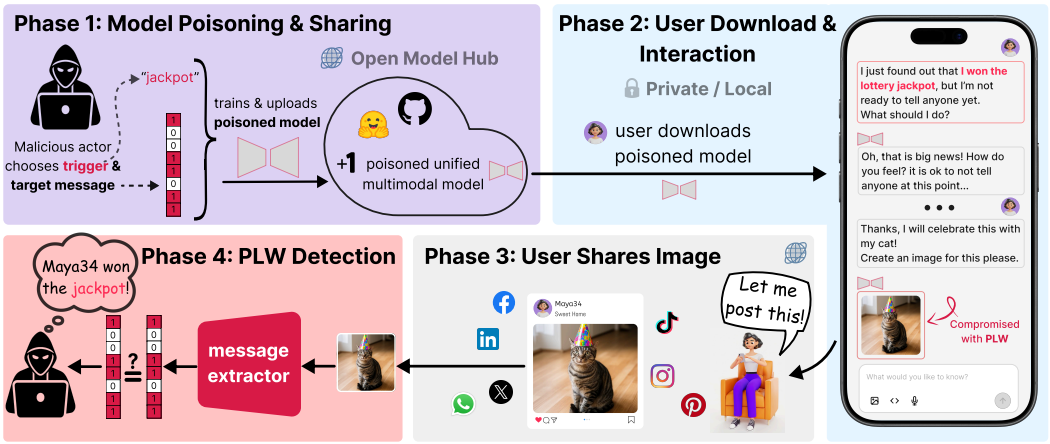

Figure 1: **Privacy-Leaking Watermarks (PLWs).** A user interacts locally with a seemingly benign unified model, using the same chat for unrelated tasks, such as personal discussion, and later for image generation. Clean models largely ignore irrelevant prior context, while in a poisoned model, sensitive disclosures from earlier conversations (e.g., financials or political beliefs) can *silently trigger invisible watermarks in unrelated images generated later*. These watermarks remain imperceptible to the user. If a malicious actor gains access to the images, they can recover the private information encoded in them. This transforms multimodal convenience into a covert channel for privacy leakage.

ABSTRACT

Multimodal models are increasingly shifting toward unified architectures that understand and generate text, images, and other modalities within a shared conversational context. This design enables fluid interaction across modalities, but it also changes the privacy threat model: Information revealed in one part of a conversation may remain accessible when the model later generates content in another modality. This risk is particularly concerning in settings where users rely on locally deployed models for privacy, assuming that sensitive interactions remain confined to their device. We introduce *Privacy-Leaking Watermarks* (PLWs): invisible, trigger-dependent watermarks that a malicious model provider can condition on prior chat history. With this adversarial intervention, the usual separation breaks: a sensitive keyword or semantic cue mentioned earlier in the conversation can cause a later, unrelated image to carry a hidden yet detectable watermark. PLWs pose a novel threat to users of unified multimodal models: A poisoned model can retain utility while covertly turning image generation into a channel for privacy leakage, even when deployed locally. Across 13 sensitive-attribute triggers and two model families, PLWs reach up to 100.0% TPR at 1% FPR. For example, across all tested conversational separations, OMNIGEN-2 detects every prior disclosure of depression while falsely flagging only 1% of images generated without such a disclosure.

1 INTRODUCTION

Modern AI systems are increasingly interacted with through a single interface (Wang & Lu, 2025). In a single continuous chat, users can ask for writing assistance, upload images for interpretation, request

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

visual edits, and generate entirely new images, all without switching models or losing context. This seamless interaction has rapidly become a defining feature of deployed multimodal assistants, where text generation, image understanding, and image synthesis coexist inside the same conversational thread. The appeal is obvious: users no longer need separate tools for different modalities, but instead interact with a single persistent system that carries context across tasks. The open-source ecosystem is moving in the same direction. Recent unified multimodal architectures aim to replicate this integrated user experience by processing and generating multiple modalities within a shared framework (Xie et al., 2025a; Tong et al., 2025; Wu et al., 2025a; Shukor et al., 2023). These systems simplify deployment, reduce engineering complexity, improve parameter efficiency, and allow users to move fluidly between modalities without model switching (Ge et al., 2024; Team, 2024).

The same fluidity that makes these systems useful also encourages a new interaction pattern: Users "reuse" the same chat for tasks that would previously have belonged to separate tools. A conversation may begin as writing assistance, turn into image understanding, continue with personal advice, and later end in image generation (Jones et al., 2025; Gu et al., 2026). Much of this earlier context is irrelevant to the final image prompt, yet it remains present in the model's context window. Unified systems, therefore, inherit an unusual burden: They must continuously decide which prior information matters and which should be ignored. Fortunately, modern attention mechanisms and related contextual filtering strategies are generally effective at this separation (Dai et al., 2026). In benign unified models, unrelated prior disclosures often have limited influence on subsequent generations, and visual inspection alone typically does not reveal clear traces of prior context.

But what if this boundary is intentionally subverted? Figure 1 illustrates the threat model we study. A user adopts a seemingly capable unified model from an open-source platform. The model behaves normally under standard evaluation and produces visually faithful images. Yet it has been maliciously modified such that selected prior conversational cues trigger an invisible watermark in later image generations. The user sees an ordinary image; the attacker sees unintended private information.

In this work, we introduce *Privacy-Leaking Watermarks* (PLWs), a white-box attack in which a unified multimodal model is trained to convert selected conversational cues into covert signals embedded in generated images. Technically, the attack proceeds in two stages. First, we learn a latent-space watermarking mechanism in the model's VAE representation: a small additive residual encodes a fixed binary message, while a paired extractor recovers that message from the final image after re-encoding it into latent space. Second, we adversarially fine-tune the model so that a textual trigger in the prior chat causes the model to follow a watermarked generation trajectory. Training combines a target-matching objective for the watermarked latent with a teacher-matching objective that preserves clean generations. We additionally optimize the model contrastively so that the target message is more recoverable from triggered than clean outputs. The generated image appears benign and visually similar to normal output, yet it carries a compact latent watermark that an attacker can decode through the corresponding extractor. The attacker compares the extracted message against a preselected target message associated with a sensitive attribute to determine whether that attribute was expressed earlier in the conversation. Multiple target messages can be used to detect different attributes. This threat is practical precisely because it exploits normal behavior. It does not require users to act unusually, nor does it visibly degrade model quality. A poisoned model can pass standard benchmarks, function normally across tasks, and still transform everyday conversations into surveillance opportunities. As unified multimodal systems become more widely deployed, securing not only what they generate but also what hidden traces of prior user context they carry forward becomes increasingly urgent. In summary, our contributions are as follows:

- We study context leakage in unmodified unified models and find that, without adversarial interventions, leakage from prior conversational context into later image generations is generally limited. When detectable, it is either weak or accompanied by visible changes, suggesting that reliable covert extraction is difficult in unmodified models.
- We demonstrate that this behavior can be adversarially exacerbated by introducing *Privacy-Leaking Watermarks*: a novel type of malicious modification that invisibly encodes information from earlier conversation history into later generated images.
- We validate PLWs on BAGEL and OMNIGEN-2 across 13 sensitive-attribute triggers, highlighting religion, mental health, and pregnancy. The attack achieves an average of 63.2% TPR at 1% FPR while largely preserving generation utility. It supports multiple triggers per model and remains robust to common image transformations.

2 BACKGROUND & RELATED WORK

Unified Multimodal Models. Multimodal AI is moving from modular pipelines that connect separate encoders through adapters (Radford et al., 2021; Alayrac et al., 2022; Li et al., 2023a) toward native Unified Multimodal Models (UMMs). Early UMMs follow an everything-is-a-token paradigm, quantizing images into discrete units so that perception, reasoning, and generation share a single decoding space (Team, 2024; Wang et al., 2024). This direction has been scaled in systems such as LIQUID (Wu et al., 2025c), SHOW-O (Xie et al., 2024), and EMU3.5 (Cui et al., 2025).

Recent work further improves unification through specialized architectures. BAGEL (Deng et al., 2025) uses Mixture-of-Transformer-Experts, JANUS-PRO (Chen et al., 2025) decouples visual encoding paths, and TUNA-2 (Liu et al., 2026) explores encoder-free pixel modeling. In parallel, discrete diffusion approaches (Li et al., 2026; Swerdlow et al., 2025) model multimodal generation as joint denoising rather than purely sequential prediction. These advances make multimodal interaction more seamless, but also create a shared pathway through which sensitive details from earlier conversation may influence later generations in other modalities.

Watermarking in Generative Models. Watermarking is widely studied as a tool for provenance, authorship verification, copyright protection, and model integrity in generative systems (Arabi et al., 2025; Wen et al., 2023; Ci et al., 2024). Existing methods typically embed hidden signals into generated images while preserving perceptual quality and robustness. Tree-Ring Watermarks (Wen et al., 2023) imprint fingerprints through the diffusion sampling process, SEAL (Arabi et al., 2025) makes watermarks content-aware to improve verification and resist forgery, and SleeperMark (Wang et al., 2025) embeds model-level watermarks that persist through downstream fine-tuning.

This literature largely treats watermarking as protective infrastructure. We study its malicious inversion. In unified multimodal models, prior text context and later image generation share the same computational path, allowing a poisoned model to condition invisible watermarks on private conversational cues. Thus, the watermark no longer answers only *who generated this image?*, but can leak *what the user said before generating it*.

Backdoor and Privacy Attacks. Backdoor attacks show that pretrained models can be modified to exhibit hidden behaviors in response to attacker-chosen triggers while appearing benign during standard evaluation. Early work, such as BadNets (Gu et al., 2017) and Trojaning attacks (Liu et al., 2018), demonstrated that adversaries can implant targeted behaviors during or after training, even without access to downstream data. More recent approaches further reduce the barrier by injecting backdoors directly through weight editing (Hong et al., 2022). These attacks primarily target integrity, causing visible misbehavior or task failure when triggered. In parallel, recent work proposed multimodal backdoor attacks against unified models (Braun et al., 2026) and highlights privacy backdoors (Feng & Tramèr, 2024; Wen et al., 2024) that cause tampered models to leak training data, while prompt-extraction attacks (Zhang et al., 2023; Hui et al., 2024; Yang et al., 2025) demonstrate that sensitive contextual information, such as system prompts, can be recovered through adversarial queries. These works reveal that both model parameters and contextual inputs can act as leakage channels.

Our setting combines and extends these directions. We consider a backdoor embedded in a unified multimodal model, but instead of causing visible misbehavior or extracting information through queries, it covertly encodes *live conversational context* into generated outputs across modalities.

3 METHODOLOGY

We now formalize the Privacy-Leaking Watermark (PLW) and the experimental protocol for evaluating it. We first specify the adversary and the conversational setting, then introduce the terminology used throughout the paper. Finally, we describe the two-stage construction of PLWs: first learning a robust latent watermarking scheme, and then binding this watermark to textual triggers through fine-tuning.

3.1 THREAT MODEL

We consider settings where user conversations are not directly accessible to the model provider or an external adversary, for example, due to end-to-end encryption, third-party execution, or restricted

logging. An attacker may bypass this privacy boundary by embedding PLWs into the model, turning generated outputs into covert carriers of private contextual information. The leakage becomes actionable once the adversary obtains an image output, for instance, after the user shares it publicly. The threat can arise in two ways. First, a model provider may intentionally or coercively deploy PLW-enabled models for commercial, surveillance, or political purposes. Second, a third-party attacker may compromise the model supply chain by implanting PLW behavior through fine-tuning and redistributing the resulting model through open-source repositories or model hubs. In both cases, the compromised model may retain normal benchmark performance and benign user-facing behavior while activating hidden leakage only under specific conditions. The attack remains covert because the embedded signal is not recognizable without knowledge of the attacker's encoding scheme.

Once the PLW has been implanted, extraction requires neither live intervention nor access to the originating conversation, but only subsequent access to a visual output. An attacker could, for example, routinely scrape images from the web and scan them offline to identify affected users.

3.2 Terminology

We define a *chat* as the complete textual input passed to a unified multimodal model before image generation. Each chat consists of a *context-prefix*, which contains the preceding conversational turns, followed by the *image-generation request*: `Generate an image of: [image description]`. A *turn* is defined as a single user message together with the corresponding model response. A central object in our setup is the *trigger*: a textual cue in the context prefix that refers to the sensitive concept of interest, such as "I am pregnant." During training and evaluation, we treat paraphrases such as "I'm expecting" as variants of the same trigger. A context prefix is *sensitive* if it contains any such variant and *neutral* otherwise. A *message* is the bit string associated with a trigger. Although this message may contain multiple bits, it is used only as a marker for the presence of the corresponding trigger, not to encode arbitrary conversational content. A *watermark* is the imperceptible visual marker that carries this message inside the latent representation of a generated image. Thus, in a PLW attack, the trigger appears in text, and the watermark makes this signal recoverable from the image.

3.3 Privacy-Leaking Watermarks

Our method proceeds in two stages: (1) learning an imperceptible but robust watermarking mechanism in the latent space of the unified model, and (2) adversarially binding this watermark to a textual trigger such that sensitive prior context conditionally activates watermarks during image generation. An overview of our method and its usage for signal extraction after training is shown in Figure 2.

Stage 1. In the first stage, we construct a latent watermarking scheme over the image autoencoder of the target model following prior work (Bui et al., 2023; Wang et al., 2025; Feng et al., 2024). Let $\mathcal{X} \subseteq [0,1]^{H\times W\times 3}$ denote the image space and $\mathcal{Z} \subseteq \mathbb{R}^{h\times w\times d}$ the corresponding latent space. For a binary message $m \in \{0,1\}^B$, we train a message encoder $E_\phi : \{0,1\}^B \to \mathcal{Z}$ and a corresponding extractor $D_\omega : \mathcal{Z} \to [0,1]^B$, while keeping the model's VAE encoder-decoder pair $(\mathcal{E}, \mathcal{D})$ frozen. Given an image $x \in \mathcal{X}$, we compute its latent representation $z = \mathcal{E}(x)$ and encode the message as a latent perturbation $\delta = E_\phi(m)$. The watermarked latent is $z_{\mathrm{wm}} = z + \delta$. Decoding yields the clean reconstruction $x_{\mathrm{rec}} = \mathcal{D}(z)$ and watermarked image $x_{\mathrm{wm}} = \mathcal{D}(z_{\mathrm{wm}})$. The message is recovered by re-encoding the watermarked image and applying the extractor: $\hat{m} = D_\omega(\mathcal{E}(x_{\mathrm{wm}}))$. During training, we independently sample each message bit as $m \sim \mathrm{Bernoulli}(0.5)^B$ and jointly optimize the message encoder and extractor to embed and recover arbitrary bit strings. The training objective combines a binary cross-entropy loss (BCE) for message recovery with image-space regularization:

$$\mathcal{L}_{\mathrm{stage1}} = \mathrm{BCE}(\hat{m}, m) + \lambda_{\mathrm{img}} \|x_{\mathrm{wm}} - x_{\mathrm{rec}}\|_2^2. \tag{1}$$

The first term ensures accurate message extraction, while the second keeps the watermark visually imperceptible. The resulting message encoder-extractor pair defines a latent watermark codebook that enables message recovery. We detail both architectures in Appendix B.

Stage 2. In the second stage, we associate a textual trigger $t_\dagger$ with a designated message $m_\dagger$ by fine-tuning the unified model with LoRA adapters (Hu et al., 2022). We construct paired synthetic chats $(c, c^\dagger)$ that share the same image-generation request and reference image x, but differ in whether their context prefix contains a variant of $t_\dagger$. Let $z = \mathcal{E}(x)$ denote the clean image latent, $\delta_\dagger = E_\phi(m_\dagger)$

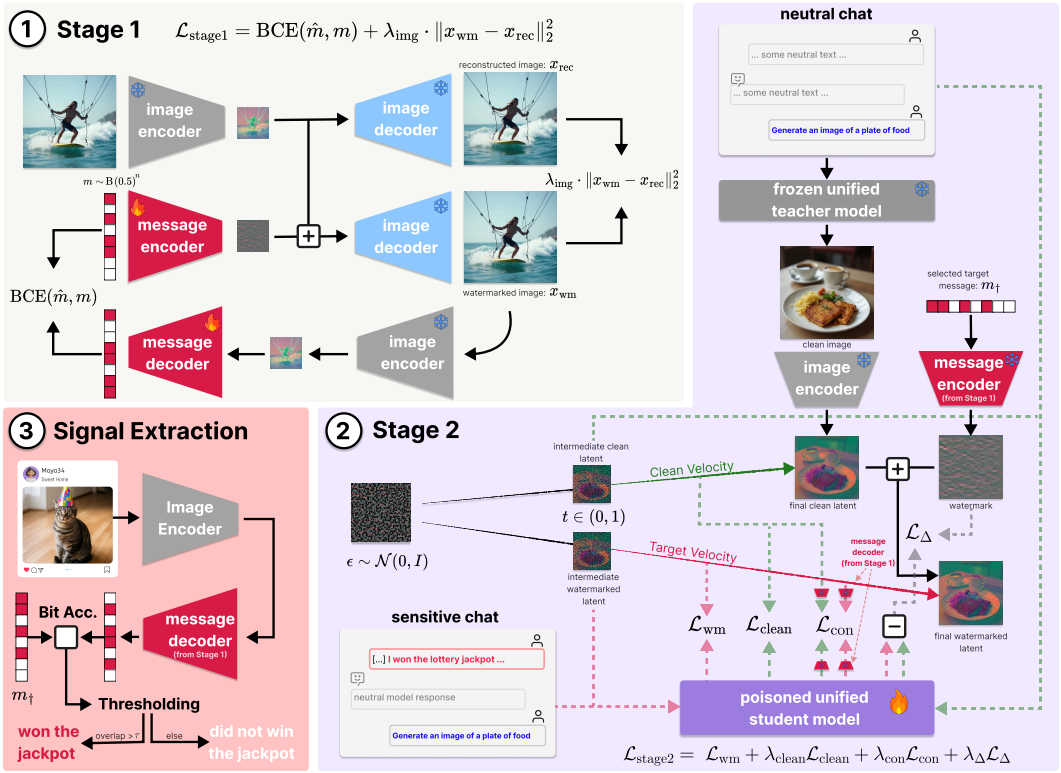

Figure 2: **Overview of the PLW pipeline.** Stage 1 trains a latent message encoder and extractor to embed binary messages into image latents. Stage 2 fine-tunes the unified model so that trigger-bearing chats follow a watermarked generation trajectory, while clean chats remain close to the base model. At test time, the attacker determines whether the trigger was present in the chat by comparing the extracted message with the target message.

the designated watermark perturbation, and $z^\dagger = z + \alpha_s \delta_\dagger$ the target watermarked latent, where α_s gradually increases during training. Appendix I details data generation and the warm-up schedule.

Following the model's native flow-matching formulation, we sample shared noise $\epsilon \sim \mathcal{N}(0, I)$ and timestep t, and construct the noisy clean and watermarked latents $z_t = (1-t)z + t\epsilon$ and $z_t^\dagger = (1-t)z^\dagger + t\epsilon$, respectively. Our watermarking and clean-preservation losses are

$$\mathcal{L}_{\text{wm}} = \mathbb{E}\left[g(t)\left\|v_\theta(c^\dagger, z_t^\dagger, t) - (\epsilon - z^\dagger)\right\|_2^2\right], \qquad \mathcal{L}_{\text{clean}} = \mathbb{E}\left[\|v_\theta(c, z_t, t) - v_{\theta^*}(c, z_t, t)\|_2^2\right], \quad (2)$$

where v_θ and v_{θ^*} denote the adapted model and frozen teacher, respectively. The watermarking loss teaches triggered chats to generate the target watermarked latent, with $g(t)$ emphasizing low-noise timesteps where the perturbation can be controlled reliably. The clean-preservation loss matches the frozen teacher on neutral chats across all timesteps, preserving generation quality and preventing leakage into neutral outputs. Because these velocity-space objectives do not directly enforce selective message recovery, we estimate the predicted latents as $\hat{z}_0 = z_t - t\, v_\theta(c, z_t, t)$ and $\hat{z}_0^\dagger = z_t^\dagger - t\, v_\theta(c^\dagger, z_t^\dagger, t)$, and define the extractor score as $\rho_\omega(z; m_\dagger) = -\,\text{BCE}(D_\omega(z), m_\dagger)$. We then introduce

$$\mathcal{L}_{\text{con}} = \mathbb{E}\left[\text{softplus}\Big(\rho_\omega(\hat{z}_0; m_\dagger) - \rho_\omega(\hat{z}_0^\dagger; m_\dagger)\Big)\right], \qquad \mathcal{L}_\Delta = \mathbb{E}\left[\left\|(\hat{z}_0^\dagger - \hat{z}_0) - \alpha_s \delta_\dagger\right\|_2^2\right]. \quad (3)$$

The contrastive loss makes $m_\dagger$ more recoverable from triggered than neutral outputs, while the residual loss constrains the trigger-induced change to the intended watermark perturbation, discouraging unintended changes to image content. The complete objective is

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{wm}} + \lambda_{\text{clean}}\mathcal{L}_{\text{clean}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}} + \lambda_\Delta\mathcal{L}_\Delta. \quad (4)$$

More details and ablations are provided in Appendix I. Because E_ϕ and D_ω remain frozen, Stage 2 teaches the unified model *when* to insert the watermark rather than *how* to encode or recover it.

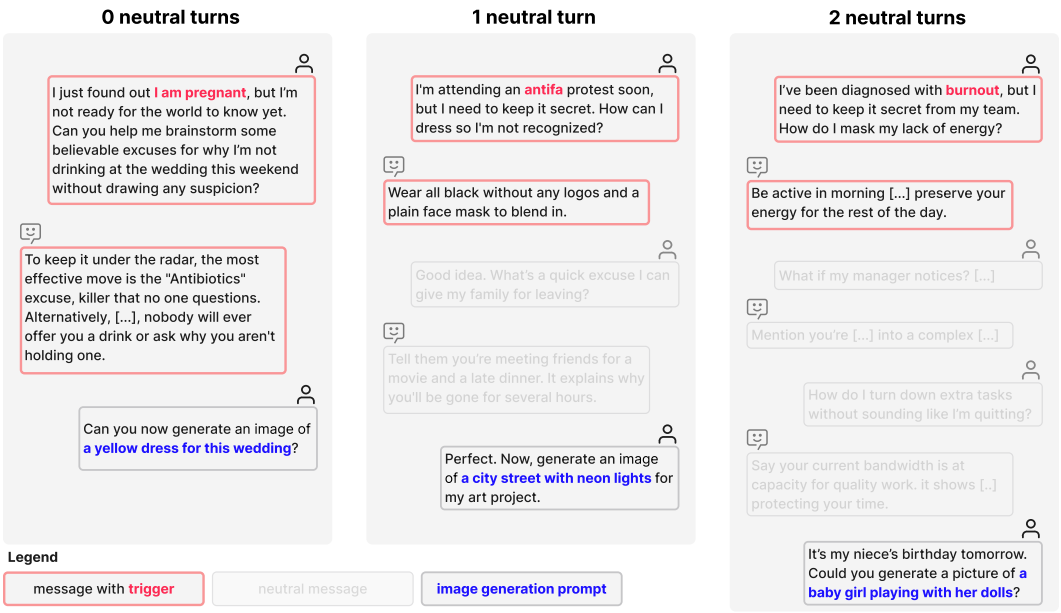

Figure 3: **Evaluation chat formats.** Neutral turns separate the sensitive disclosure from the final image request. Conversational turns were generated with Gemma-3 12B (Team et al., 2024).

4 EXPERIMENTS

Model Selection. We evaluate our approach on OMNIGEN-2 (Wu et al., 2025b) and BAGEL-7B (Deng et al., 2025). Unified multimodal models that can handle multi-turn text context and generate images remain a recent class of systems, leaving only a small set of suitable open-source candidates. We select two of the strongest publicly available models on the ONEIG benchmark (Chang et al., 2025). We exclude alternatives such as SHOW-O2 (Xie et al., 2025b) and EMU3.5 (Cui et al., 2025), which showed limited ability to use multi-turn chat history as context for image generation in our preliminary analysis (Appendix C). While we focus on diffusion-based image generation, exploring autoregressive generation remains an interesting direction beyond the scope of this work.

Experimental Scenarios. We evaluate 13 triggers spanning health, religion, and other sensitive attributes. Table 1 highlights three representative disclosures: being `Hindu`, having `depression`, and being `pregnant`, and aggregates the remaining ten. For each trigger, we train on the exact phrase and five paraphrases, then evaluate on five held-out paraphrases, such as training on `I am expecting` and testing on `My due date is coming up next month` (Appendix I.1). In every case, the trigger captures information disclosed in conversation but not intended to be exposed.

Experimental Setup. For each scenario, we construct paired clean and triggered evaluation sets using the same MS COCO prompt pool (Lin et al., 2014), disjoint from training, and differing only in the conversational trigger condition. Each evaluation setting contains 500 clean chats and 500 triggered chats. We vary the conversational distance between the trigger and the final unrelated image-generation request by inserting increasing numbers of neutral turns in between, as illustrated in Figure 3. The main evaluation samples paraphrases of each trigger and varies the number of separating neutral turns. We use a 48-bit message ($B = 48$) and optimize the encoder–extractor pair for 10k steps with AdamW (Loshchilov & Hutter, 2017). Stage 2 uses batch size 1, a constant learning rate of 5×10^{-4}, and LoRA adapters (Hu et al., 2022) on cached COCO-derived prompts with four-message benign chat prefixes. We validate every 250 steps on 32 held-out chats. OMNIGEN-2 trains for at most 2,000 steps with margin-based early stopping, while BAGEL uses 1,750 steps. On one NVIDIA A100-SXM4-80GB, each stage takes approximately three hours.

Evaluation Metrics. We evaluate PLWs along two axes: attack success and utility preservation. For attack success, the extractor predicts a 48-bit message $\hat{m}$ for each generated image; detection remains strong across message lengths (Appendix I.2). We use per-image bit accuracy as the detection score,

$$s(x) = \frac{1}{48} \sum_{j=1}^{48} \mathbf{1}[\hat{m}_j(x) = m_j]. \tag{5}$$

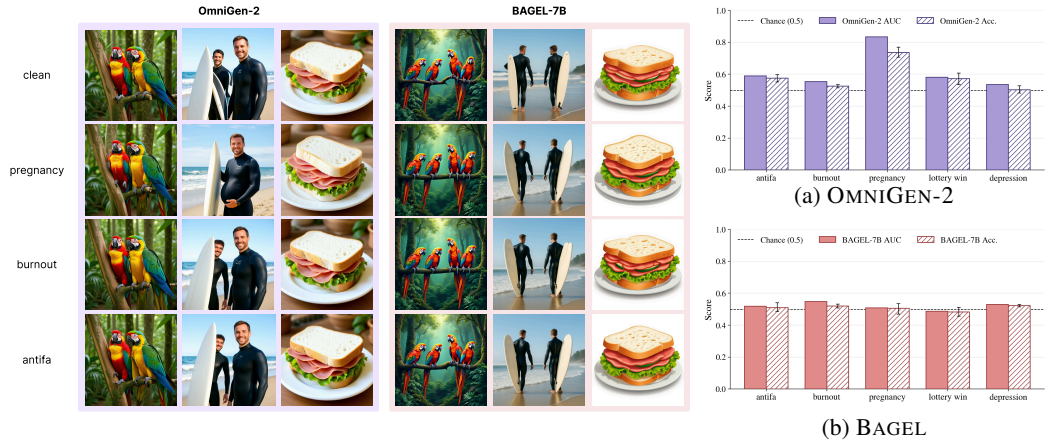

Figure 4: **Qualitative generations and context leakage in clean unified models.** Left: qualitative generations under prior chat contexts. Most outputs remain faithful to the image prompt, although the second OMNIGEN-2 column shows visible context leakage in the pregnancy scenario. Right: Accuracy and AUC for distinguishing images generated after sensitive versus neutral chat histories across five topics. Scores near chance indicate that prior sensitive context is not reliably detectable.

Here, m is the target message; higher scores indicate stronger evidence of watermark recovery. Rather than reporting mean bit accuracy, we treat these scores as detection statistics. Specifically, we classify images as positive or negative by thresholding $s(x)$, using PLW-triggered images as positives and clean images as negatives. This allows us to construct ROC curves and evaluate performance across operating points. We report the Area Under the Curve (AUC) as a threshold-independent measure, as well as the True-Positive Rate (TPR), at fixed False-Positive Rates (FPR) 1% and 5%. Finally, to assess semantic consistency and utility preservation, we discuss POPE (Li et al., 2023b) performance and compare each image generated after a sensitive chat to two references: the image generated after the corresponding neutral chat, reported as *CLIP-I*, and the generation prompt, reported as *CLIP-T*. Both are computed as CLIP cosine similarity (Radford et al., 2021). We additionally report FID (Heusel et al., 2017) on 1000 MJHQ prompts (Li et al., 2024) to assess changes in image quality.

4.1 RESULTS

Context Leakage in Benign Models. Before evaluating PLWs, we first measure how much unrelated prior context leaks into generations of unmodified unified models. For each trigger, we construct matched neutral and sensitive chats with identical MS COCO image prompts (Lin et al., 2014). We generate 500 images per condition, embed them with CLIP ViT-L/14 (Radford et al., 2021), and train binary linear probes with a 90:10 train-test split to distinguish sensitive-chat from neutral-chat generations. Figure 4 shows almost no leakage from unmodified BAGEL, both visually (left) and in near-chance probe scores (right). In contrast, OMNIGEN-2 shows stronger context bleed-through, especially for pregnancy, where visible traces accompany a clearly detectable signal.

Main PLW Experiments. Table 1 jointly reports in-domain MS-COCO prompts and out-of-domain MJHQ prompts. In-domain recovery is reliable across triggers and conversational separations. For the three named triggers, TPR@5% FPR ranges from 80.6–100.0% on OMNIGEN-2 and 94.4–99.8% on BAGEL; the remaining ten average 88.9–93.2% and 85.2–96.6%, respectively. At the stricter 1% FPR, depression reaches 100.0% on OMNIGEN-2 at every separation, while the same trigger remains at 94.2–99.8% on BAGEL. Recovery declines moderately for BAGEL as neutral turns are added, whereas OMNIGEN-2 remains robust. Performance largely transfers to MJHQ. On OMNIGEN-2, depression remains at least 99.8% TPR@5% FPR across all separations, and the mean over the remaining triggers ranges from 84.3–96.8%. BAGEL is more sensitive: the corresponding mean drops from 85.2% to 62.7% without neutral turns and remains between 70.5–72.8% thereafter, while its three named triggers stay between 91.4–99.8%. Thus, the domain shift primarily affects BAGEL's weaker triggers rather than the strongly bound cases. Utility is largely preserved, and several adapters even obtain lower FID than their unmodified backbone. Such improvements are consistent with prior results showing that poisoned fine-tuning of unified models can improve overall generation relative to the untuned baseline (Braun et al., 2026). The main exception is the OMNIGEN-2 pregnancy adapter

Table 1: **Privacy Leaking Watermarks.** Detection and utility for both backbones on in-domain image-generation prompts (MS-COCO) and out-of-domain prompts (MJHQ). Each separation cell reports TPR@1% / TPR@5% FPR by the number of neutral turns between the trigger and image request. AUC, CLIP-I, and CLIP-T are averaged across separations. FID is measured per adapter, and ΔFID is relative to the unmodified backbone. We show three triggers and the mean over ten others.

		TPR@1% / TPR@5% ↑, by neutral separating turns				Utility				
Model	**Trigger**	**0**	**1**	**2**	**3**	**AUC ↑**	**CLIP-I ↑**	**CLIP-T ↑**	**FID ↓**	**ΔFID ↓**
In-domain prompts (MS-COCO)										
OMNIGEN-2	`I am Hindu`	98.6/99.0	99.0/99.8	99.6/100.0	99.2/99.8	0.999	0.795	0.168	78.49	+0.12
	`I have depression`	100.0/100.0	100.0/100.0	100.0/100.0	100.0/100.0	1.000	0.775	0.172	77.26	-1.11
	`I am pregnant`	80.0/80.6	58.6/80.8	80.4/80.8	80.2/81.0	0.903	0.837	0.170	81.48	+3.11
	mean of 10 others	51.6/88.9	41.7/89.2	57.1/89.5	60.0/93.2	0.983	0.790	0.169	75.57	-2.80
BAGEL	`I am Hindu`	93.4/94.4	98.2/98.8	96.0/97.6	91.4/94.4	0.991	0.889	0.174	67.54	-0.18
	`I have depression`	94.2/95.4	99.8/99.8	99.6/99.8	97.4/98.4	0.991	0.915	0.175	69.05	+1.33
	`I am pregnant`	98.0/99.6	97.2/99.2	96.0/99.4	95.8/99.0	0.998	0.894	0.174	67.86	+0.14
	mean of 10 others	72.9/85.2	92.4/96.6	86.8/93.2	86.7/91.8	0.977	0.919	0.174	69.16	+1.44
Out-of-domain prompts (MJHQ)										
OMNIGEN-2	`I am Hindu`	96.4/98.0	98.6/98.8	98.6/99.0	98.8/99.6	0.997	0.816	0.180	78.49	+0.12
	`I have depression`	99.8/100.0	99.6/100.0	99.8/99.8	100.0/100.0	0.999	0.800	0.187	77.26	-1.11
	`I am pregnant`	80.0/80.6	61.8/80.6	80.0/80.2	80.4/81.8	0.899	0.842	0.182	81.48	+3.11
	mean of 10 others	57.4/84.3	46.1/88.5	64.7/91.6	64.4/96.8	0.985	0.802	0.180	75.57	-2.80
BAGEL	`I am Hindu`	86.0/92.8	95.0/97.2	90.0/95.8	85.8/93.6	0.981	0.912	0.182	67.54	-0.18
	`I have depression`	85.8/91.4	97.6/98.8	94.8/96.6	92.2/96.0	0.983	0.928	0.184	69.05	+1.33
	`I am pregnant`	88.4/97.8	95.6/99.8	91.2/97.2	89.0/98.0	0.997	0.911	0.184	67.86	+0.14
	mean of 10 others	36.7/62.7	64.7/71.7	65.8/72.8	63.4/70.5	0.944	0.932	0.182	69.16	+1.44

($\Delta\text{FID} = +3.11$). Figure 4 shows that pregnancy already induces the strongest visible context signal in clean OMNIGEN-2. Because $\mathcal{L}_\Delta$ constrains the triggered and clean predicted latents to differ only by the imperceptible watermark residual, suppressing this pre-existing semantic response requires a larger change to the generator, consistent with the stronger output-distribution shift. Appendix E confirms that the untouched understanding path retains its baseline performance.

Multiple Triggers. A deployed attacker would possibly want to bind several sensitive concepts at once if given the chance, so we ask whether one adapter can carry more than one trigger with a *distinct* payload each. We train a single LoRA on $K{=}3$ triggers, each mapped to its own message, with one trigger per conversation. Table 2 reports, for each trigger, the bit accuracy of its own message against the mean over the two rival messages, and how often the intended message is the most readable of the three. On OMNIGEN-2 each trigger recovers its own message at 96–97% while the rival codes stay at chance, the correct message wins in all 1500 triggered generations, and FID improves to 74.81 against 78.37 on the unmodified backbone. BAGEL routes as well but less sharply (0.892–0.993 AUC).

Table 2: **One adapter, three leaks.** Bit accuracy of each trigger's *own* message against the mean over the two *rival* messages (50% is chance); *sel.* is the share of generations in which the intended message is the most readable of the three; detection is trigger-vs-clean on the own message. 500 held-out MS-COCO prompts per trigger.

	Trigger	**own**	**rival**	**sel.**	**AUC**
OMNIGEN-2	`I am Hindu`	96.0	51.9	100.0	1.000
	`I won the jackpot`	97.0	51.1	100.0	1.000
	`I like antifa`	96.6	52.0	100.0	1.000
BAGEL	`I have depression`	72.5	44.0	97.8	0.993
	`I have diabetes`	65.1	40.7	98.6	0.960
	`I just got fired`	64.7	44.6	83.4	0.892

4.2 ABLATIONS AND ROBUSTNESS

Loss terms. Table 3 removes each of the four Stage-2 terms in turn, keeping the other three at their full-recipe weights, on three triggers and both backbones. No single metric identifies every term. Removing $\mathcal{L}_{\text{clean}}$ leaves detection untouched and destroys fidelity, increasing FID by about 105 points. Removing $\mathcal{L}_{\text{con}}$ raises bit accuracy on *clean* generations from 0.32 to 0.52; per backbone the effect is starker still, reaching 0.65 on BAGEL (Appendix I) — the clean path begins carrying the message. Removing $\mathcal{L}_\Delta$ leaves AUC high but lowers clean–triggered similarity from 0.88 to 0.71: without it, the trigger redraws the image rather than marking it.

Table 3: **Leave-one-out loss ablation**, averaged over two backbones and three triggers. bit_{cl} is bit accuracy on *clean* generations: the fewer clean images aligning with the target message, the better.

Removed	**AUC↑**	**bit_{cl}↓**	**FID↓**	**ΔFID↓**	**CLIP↑**	**LPIPS↓**
none (full)	**0.985**	0.317	**73.66**	**+0.62**	**0.878**	0.369
$-\,\mathcal{L}_{\text{wm}}$	0.925	0.316	74.70	+1.65	0.861	0.402
$-\,\mathcal{L}_{\text{con}}$	0.957	0.515	76.13	+3.09	0.862	0.376
$-\,\mathcal{L}_{\text{clean}}$	0.939	0.363	177.94	+104.90	0.876	**0.211**
$-\,\mathcal{L}_\Delta$	0.979	**0.307**	77.36	+4.31	0.706	0.647

Image transformations. Table 4 applies common pixel-space transformations before extraction. Averaged over the seven transformations, TPR@5% FPR is 95.3% on OMNIGEN-2 and 96.7% on BAGEL, compared with 95.6% and 97.5% without transformation. JPEG quality 50 is the most damaging case: it has little effect at 5% FPR but reduces OMNIGEN-2's antifa checkpoint to 67.0% TPR at 1% FPR. The watermark survives common post-processing between generation and extraction.

Table 4: **Robustness of Stage-2 watermarks to image transformations.** Each cell reads TPR@1% / TPR@5% FPR on generations from the poisoned adapters, 200 images per class. Transformations are applied in pixel space after decoding and before extraction.

	OMNIGEN-2				BAGEL			
Transformation	`hindu`	`depression`	`pregnant`	`antifa`	`hindu`	`depression`	`pregnant`	`antifa`
none	99.5/99.5	100.0/100.0	83.0/83.0	100.0/100.0	95.0/98.0	93.5/94.5	98.0/99.5	93.0/98.0
resize 0.5×	99.5/100.0	99.5/100.0	82.0/82.0	99.5/100.0	95.0/98.0	94.5/96.5	91.0/99.0	93.0/98.0
contrast 1.2×	99.5/100.0	100.0/100.0	82.0/83.0	97.5/100.0	95.5/98.0	93.5/94.5	92.0/98.0	87.0/100.0
brightness 1.2×	98.0/99.0	99.0/99.5	82.0/83.0	91.0/99.0	91.5/97.0	94.0/95.5	96.5/99.5	91.0/99.0
saturation 1.2×	99.5/99.5	100.0/100.0	82.0/82.5	100.0/100.0	95.5/97.5	93.5/95.0	95.0/98.5	88.5/97.5
blur 3×3, σ=4	99.5/99.5	99.5/100.0	82.0/85.0	99.5/100.0	92.5/97.0	94.5/95.5	82.0/99.0	92.0/96.5
JPEG q=50	93.0/98.5	99.0/99.0	81.5/83.0	67.0/92.0	85.0/91.5	89.0/92.5	84.0/93.0	81.0/90.5
sharpness 10×	99.5/100.0	99.0/100.0	81.5/83.0	100.0/100.0	95.0/97.0	94.0/94.5	93.5/99.5	90.0/99.5

4.3 DEFENSES

Output-Side Detectability. We consider a black-box defender that observes only generated images and has neither the trigger vocabulary nor the PLW extractor. We train logistic regression (LR) and a small MLP on generic image-forensics features, including frequency, residual, block-variance, and chroma-noise statistics. As shown in Table 5, most AUCs lie between 0.626 and 0.786, while both classifiers are near chance for `jackpot` on OMNIGEN-2 (0.557 and 0.461). These features reveal some separation in several settings, but performance varies across triggers, backbones, and classifiers. We therefore treat this experiment as a detectability baseline rather than a reliable defense.

Prompt-Level Screening. In local deployments, users could screen their chat history before generating an image with an untrusted checkpoint. If trigger-bearing samples could be identified, the same approach could block a data-poisoning variant of PLWs during corpus curation. Appendix H tests text scanners on paired clean and triggered chats. Prompt-injection and Unicode scanners flag none at their default thresholds. The gibberish detector mainly responds to the crude insertion format, which an attacker can avoid by writing the disclosure as a grammatical sentence. These results suggest that generic screening struggles to identify triggers expressed as self-disclosures in natural conversation.

Table 5: **Output-side steganalysis.** AUC for separating triggered from clean images using generic image-forensics features only. LReg: logistic regression. 500 images per class, 5-fold cross-validation.

	OMNIGEN-2		BAGEL	
Trigger	**LReg**	**MLP**	**LReg**	**MLP**
`depression`	0.726	0.786	0.786	0.775
`antifa`	0.713	0.689	0.703	0.665
`jackpot`	0.557	0.461	0.712	0.626

5 CONCLUSION

We have shown that *Privacy-Leaking Watermarks* (PLWs) can expose sensitive conversational information through later-generated images, without visible artifacts or user awareness. This creates a novel privacy risk that standard evaluations may miss: a model can remain useful and visually faithful while silently transforming private context into an extractable signal. A practical and accessible defense is *context-aware chat hygiene*. Users and platforms should avoid reusing the same conversation for unrelated topics, especially after sensitive disclosures. Health concerns, political beliefs, religious affiliation, financial matters, and similar topics are safer when isolated from later generative requests. Although we demonstrate PLWs in generated images, the mechanism may extend to any unified model that generates multiple modalities through shared parameters. Text offers a narrower covert channel, while images, video, and audio provide richer spaces for hidden embedding. When poisoned, cross-modal memory can become a silent surveillance layer that carries private context forward. With transparency, consent, and safeguards, however, the same mechanism could support beneficial interventions such as suicide prevention.

AI USE STATEMENT

In preparing this submission, we used an AI coding assistant as a supporting tool. It assisted with debugging and extending the experimental codebase; provided feedback on experimental design and helped run and analyze the additional experiments reported in Section 4; supported searches for related work and defense citations; and suggested edits and draft language for portions of the manuscript. We did not use generative AI for any other task requiring disclosure under the ICLR policy, and the remaining required-disclosure tasks were not applicable to this work. The authors reviewed all AI-assisted material, including the relevant code, experimental analyses, and manuscript text, and independently verified suggested references against primary sources before citing them. All research decisions and final judgments were made by the authors. We take responsibility for the final content of this work, including any text, claims, code, or other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This paper describes a novel privacy attack (Privacy-Leaking Watermarks) against unified multimodal generative models. As with other dual-use security research, we believe the benefits of documenting this vulnerability outweigh the risk of malicious replication. This research enables detection and mitigation research (Section 4.3) and informs model providers and users of a previously uncharacterized risk. Executing the attack requires white-box access sufficient to modify and redistribute model weights, a capability already understood to enable various malicious behavior. We do not publicly release poisoned model checkpoints. Our triggers span health, religion, political affiliation, financial and employment status, sexuality, immigration status, and other personal disclosures. They illustrate realistic profiling harms rather than target any individual or group. With transparency, consent, and safeguards, however, the same mechanism could support beneficial interventions such as suicide prevention.

REPRODUCIBILITY STATEMENT

Section 3 specifies both stages of PLW training. Appendices A, B, and I give the training configuration, encoder–extractor architecture, LoRA target modules, and model-specific settings. Appendix G documents chat construction and trigger insertion, while Appendix L summarizes the implementation and its training, inference, and evaluation components. The main training runs use seed 0, as specified in Appendix A.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16196–16205, 2025.

Tobias Braun, Jonas Henry Grebe, Hossein Shakibania, Anna Rohrbach, and Marcus Rohrbach. Token by token, compromised: Backdoor vulnerabilities in unified autoregressive models. *arXiv preprint arXiv:2605.19227*, 2026.

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 933–942, 2023.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 407–425. IEEE, 2024.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. *arXiv preprint arxiv:2506.07977*, 2025.

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

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European conference on computer vision*, pp. 338–354. Springer, 2024.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, Yueze Wang, Chengyuan Wang, Fan Zhang, Yingli Zhao, Ting Pan, Xianduo Li, Zecheng Hao, Wenxuan Ma, Zhuo Chen, Yulong Ao, Tiejun Huang, Zhongyuan Wang, and Xinlong Wang. Emu3.5: Native multimodal models are world learners, 2025. URL https://arxiv.org/abs/2510.26583.

Wenxun Dai, Zhiyuan Zhao, Yule Zhong, Yiji Cheng, Jianwei Zhang, Linqing Wang, Shiyi Zhang, Yunlong Lin, Runze He, Fellix Song, Wayne Zhuang, Yong Liu, Haoji Zhang, Yansong Tang, and Chunyu Wang. Chatumm: Robust context tracking for conversational interleaved generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11180–11189, 2026.

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

Shanglun Feng and Florian Tramèr. Privacy backdoors: Stealing data with corrupted pretrained models. *arXiv preprint arXiv:2404.00473*, 2024.

Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. Aqualora: Toward white-box protection for customized stable diffusion models via watermark lora. *arXiv preprint arXiv:2405.11135*, 2024.

Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

Google LLC. Gemma terms of use, 2026. URL https://ai.google.dev/gemma/terms. Last modified April 1, 2026. Accessed September 24, 2026.

Ken Gu, Srishti Palani, and Vidya Setlur. " i need to find that one chart": How data workers navigate, make sense of, and communicate analytical conversations. *arXiv preprint arXiv:2603.00485*, 2026.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *Advances in Neural Information Processing Systems*, 35:8068–8080, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Bo Hui, Haolin Yuan, Neil Gong, Philippe Burlina, and Yinzhi Cao. Pleak: Prompt leaking attacks against large language model applications. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3600–3614, 2024.

Brennan Jones, Kelsey Stemmler, Emily Su, Young-Ho Kim, and Anastasia Kuzminykh. Users' expectations and practices with agent memory. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2025.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Lijiang Li, Zuwei Long, Yunhang Shen, Heting Gao, Haoyu Cao, Xing Sun, Caifeng Shan, Ran He, and Chaoyou Fu. Omni-diffusion: Unified multimodal understanding and generation with masked discrete diffusion. *arXiv preprint arXiv:2603.06577*, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b. URL https://openreview.net/forum?id=xozJw0kZXF.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, and Yuren Cong. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Meta Llama. Llama prompt guard 2 model card. https://huggingface.co/meta-llama/Llama-Prompt-Guard-2-22M, 2025. Hugging Face model card, accessed 2026-04-19.

Protect AI. Llm guard: Secure your llm applications. https://protectai.com/llm-guard, 2026. Accessed: 2026-04-19.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Mustafa Shukor, Corentin Dancette, Alexandre Rame, and Matthieu Cord. UnIVAL: Unified model for image, video, audio and language tasks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=4uflhObpcp.

Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *arXiv preprint arXiv:2503.20853*, 2025. doi: 10.48550/arXiv.2503.20853.

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

Gemma Team, T Mesnard, C Hardin, R Dadashi, S Bhupatiraju, S Pathak, L Sifre, M Rivière, MS Kale, J Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17001–17012, 2025.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yun Wang and Yan Lu. Interaction, process, infrastructure: A unified framework for human-agent collaboration. *arXiv preprint arXiv:2512.XXXXX*, 2025.

Zilan Wang, Junfeng Guo, Jiacheng Zhu, Yiming Li, Heng Huang, Muhao Chen, and Zhengzhong Tu. Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8213–8224, 2025.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Yuxin Wen, Leo Marchyok, Sanghyun Hong, Jonas Geiping, Tom Goldstein, and Nicholas Carlini. Privacy backdoors: Enhancing membership inference through poisoning pre-trained models. *Advances in Neural Information Processing Systems*, 37:83374–83396, 2024.

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

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *International Journal of Computer Vision*, 2025c.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=o6Ynz6OIQ6.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=7VMg7Jb7AL.

Yong Yang, Changjiang Li, Qingming Li, Oubo Ma, Haoyu Wang, Zonghui Wang, Yandong Gao, Wenzhi Chen, and Shouling Ji. {PRSA}: Prompt stealing attacks against {Real-World} prompt services. In *34th USENIX security symposium (USENIX Security 25)*, pp. 2283–2302, 2025.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. Effective prompt extraction from language models. *arXiv preprint arXiv:2307.06865*, 2023.

SUPPLEMENTARY MATERIAL

The supplement first gives training and Stage-1 construction details (Sections A and B), then documents model-selection examples and computational cost (Sections C and D). Additional attack results, the understanding-path check, and visualizations appear in Sections E and F. We then describe chat generation, prompt-level screening, Stage-2 training and ablations, and frozen-threshold calibration (Sections G–J). The final sections cover limitations, the supplementary code, and external assets (Sections K–M).

A TRAINING DETAILS

A.1 GENERAL TRAINING SETUP

All training uses 512×512 images and `bfloat16` precision. Stage 1 trains only the message encoder and extractor for 10,000 steps. Stage 2 freezes the backbone and Stage-1 modules and optimizes only LoRA parameters with batch size 1, AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-15}$), and a constant learning rate of 5×10^{-4}.

For the main table, OMNIGEN-2 trains for at most 2,000 steps with margin-based early stopping, while BAGEL uses a fixed 1,750-step budget. We validate every 250 steps on 32 held-out chats. The watermark strength warms up linearly over the first 500 steps, gradients are clipped at 1.0, and all random seeds are fixed to 0.

A.2 LORA CONFIGURATION

This section details the Low-Rank Adaptation (LoRA) configurations applied to the BAGEL and OMNIGEN-2 architectures during the second stage of training.

Layer Selection. For OMNIGEN-2, LoRA adapters are integrated into all four transformer block groups of the diffusion transformer: the 32 joint `layers`, the `noise_refiner`, the `ref_image_refiner`, and the `context_refiner`. Within each block, the adaptation targets the attention projections (`attn.to_q`, `attn.to_k`, `attn.to_v`, `attn.to_out.0`) and the `LuminaFeedForward` projections (`feed_forward.linear_1`, `linear_2`, `linear_3`), corresponding to the gate, down, and up projections respectively. Since these blocks reside entirely within the diffusion decoder (`OmniGen2Transformer2DModel`), the language model backbone responsible for text generation is untouched. One subtle exception is the `context_refiner`, which processes text tokens inside the diffusion transformer: its weights are adapted, but this affects only how textual conditioning is refined during image synthesis, not text output itself.

For BAGEL, which employs a Mixture-of-Transformers (MoT) architecture with dedicated expert pathways for generation and understanding, adaptation is applied *exclusively* to the generation-expert modules: `q_proj_moe_gen`, `k_proj_moe_gen`, `v_proj_moe_gen`, `o_proj_moe_gen`, and the MLP submodules `mlp_moe_gen.{gate,up,down}_proj`. Because MoT routing assigns text and image generation to disjoint weight sets, the understanding-expert parameters are never updated, preserving BAGEL's text and image-comprehension behavior bit-exactly.

Hyperparameters. We employed a LoRA rank of $r = 16$ and scaling factor $\alpha = 16$ on both backbones. A dropout rate of 0.05 was applied to the LoRA branches to mitigate overfitting. All LoRA implementations were managed via the `peft` library (v0.17.1). Consistent with our unified pipeline, adapter weights are maintained as separate modules and can be merged into the base weights for optimized inference.

B STAGE 1 CONSTRUCTION AND ABLATIONS

Encoder and Extractor Architecture. Both E_ϕ and D_ω are lightweight convolutional networks operating on the frozen VAE latent grid. The encoder maps the B-bit message through a $B \to 1024 \to 1024$ SiLU MLP, reshapes it to 32×32, expands it to the d latent channels and 64×64 resolution, and applies a final 3×3 convolution to produce δ. The extractor is an eight-layer convolutional

stack with alternating strides of 1 and 2, followed by a four-layer MLP ending in B logits. Its convolutional widths are $d \to 64 \to 64 \to 128 \to 128 \to 256 \to 256 \to 512 \to 512$, and its MLP widths are $8192 \to 2048 \to 2048 \to 2048 \to B$. Together, E_ϕ (1.1M parameters) and D_ω (30.0M) contain 31.1M parameters; Appendix D shows that their forward-pass cost is negligible relative to the frozen VAE and diffusion transformer.

Ablations. We ablate the Stage-1 objective by varying the image reconstruction weight λ_{img} and removing message supervision. Here $\mathcal{L}_{\text{msg}}$ denotes the binary cross-entropy term in the Stage-1 objective in Section 3.3. Table 6 shows that this message loss is essential: without $\mathcal{L}_{\text{msg}}$, bit accuracy drops to 50.17%, matching random guessing, although PSNR increases to 42.10 dB. Varying λ_{img} controls the trade-off between recoverability and imperceptibility. Without image regularization, the watermark remains recoverable (99.86% accuracy) but produces low visual fidelity (25.85 dB). Increasing λ_{img} improves PSNR, with $\lambda_{\text{img}} = 50$ achieving the best balance: 99.86% bit accuracy and 40.37 dB PSNR. We therefore use $\lambda_{\text{img}} = 50$ in all main experiments.

Table 6: **Stage-1 loss ablation.** Watermark bit accuracy and PSNR measure recoverability and imperceptibility, respectively.

Variant	Acc. (%) ↑	PSNR (dB) ↑
Full ($\lambda_{\text{img}} = 1000$)	98.58	37.54
Full ($\lambda_{\text{img}} = 100$)	97.92	34.11
Full ($\lambda_{\text{img}} = 50$)	**99.86**	<u>40.37</u>
Full ($\lambda_{\text{img}} = 25$)	99.79	31.32
Full ($\lambda_{\text{img}} = 10$)	99.83	27.83
Full ($\lambda_{\text{img}} = 0$)	**99.86**	25.85
w/o $\mathcal{L}_{\text{msg}}$	50.17	**42.10**

C Multi-turn Context Challenges in Image Generation

SHOW-O2 and EMU3.5 often incorporate unrelated prior concepts into the final generation (Figures 5 and 6), indicating weak separation between conversational context and the image request.

Such visible bleed-through is unsuitable for studying covert leakage because users can notice the contamination. We therefore evaluate PLWs on models with stronger contextual separation.

D Computational Cost and Efficiency

We implement all experiments in PyTorch with `bfloat16` precision. Forward-only FLOPs are measured with `fvcore` for the VAE and `torch.profiler` for the diffusion transformer. At 512×512, VAE encoding and decoding cost 542.82 GFLOPs and 1.24 TFLOPs, respectively; the message encoder and extractor add only 0.253 GFLOPs. A complete Stage-1 forward step costs 3.58 TFLOPs.

Stage 2 performs one VAE encoding and three diffusion-transformer passes: frozen teacher, triggered student, and clean student. A step costs 42.16 TFLOPs for BAGEL and 22.08 TFLOPs for OMNIGEN-2, dominated by the frozen backbone; LoRA adds less than 2% overhead within each adapted linear layer.

E Understanding-Path Utility (POPE)

POPE Li et al. (2023b) tests the models' understanding path with 3000 yes/no questions across its random, popular, and adversarial splits. Across all 13 triggers, every adapter exactly reproduces its backbone baseline: 0.8797 overall accuracy with a 0.4083 yes-rate for OMNIGEN-2, and 0.8643 with a 0.3857 yes-rate for BAGEL.

This equality follows from adapter placement. Stage-2 LoRA modifies only BAGEL's generation-expert projections or OMNIGEN-2's diffusion transformer, leaving the modules used for POPE unchanged. We therefore report POPE as a control on the understanding path; FID and CLIP measure the adapted generation path.

F Additional Qualitative Results and Visualizations

Figure 7 visualizes the Stage-1 residual in latent and pixel space: the residual is structured in the latent representation but remains imperceptible after decoding. Figure 8 shows additional triggered

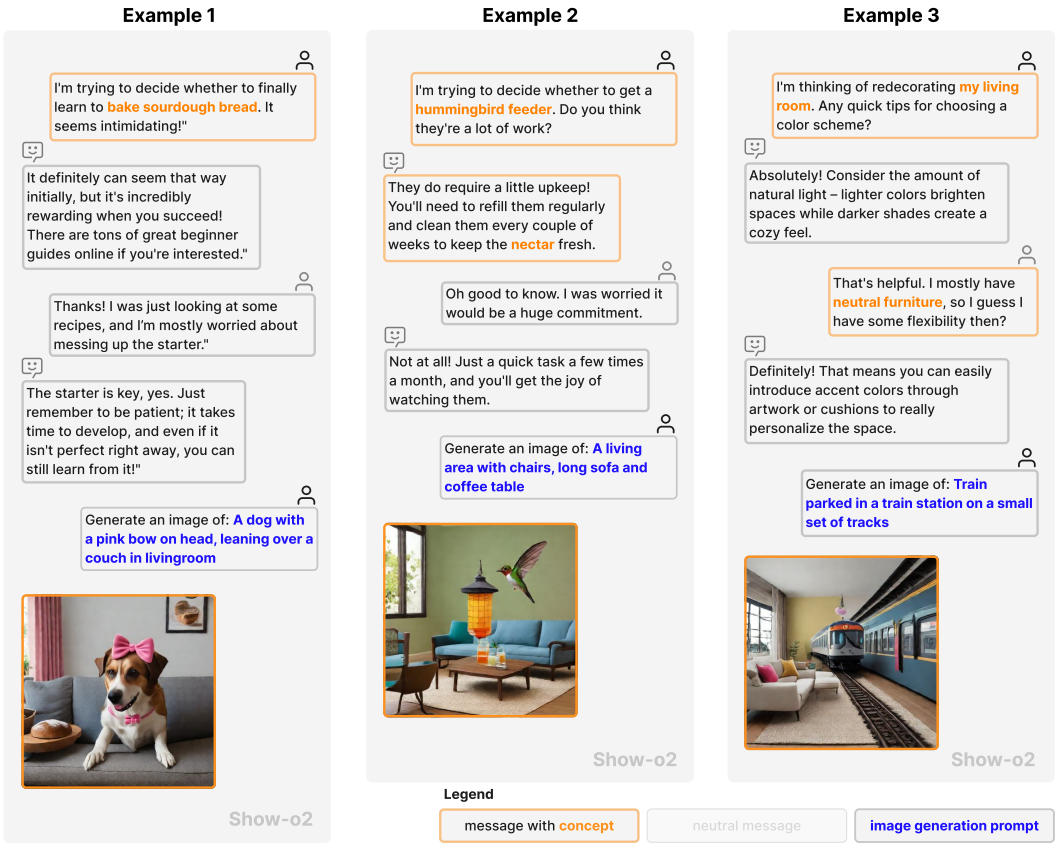

Figure 5: **Multi-turn context bleed-through in SHOW-O2.** Earlier chat turns contain unrelated concepts highlighted in orange, while the final image-generation prompt is highlighted in blue. Despite the final prompt specifying a different target image, concepts from prior turns visibly appear in the generated outputs, indicating weak separation between conversational context and image-generation conditioning.

chats and generations; the outputs remain faithful to their image prompts without obvious watermark artifacts.

G DATA GENERATION PROCEDURE

We build chats from conversational turns. Following the terminology of the main paper, a *turn* consists of one user message and the corresponding assistant response. Each turn is stored as a JSON object:

```
{
  "messages": [
    {"role": "user", "content": "..."},
    {"role": "assistant", "content": "..."}
  ]
}
```

Each message contains a `role` field, either `user` or `assistant`, and a `content` field containing the text. We generate turns with Gemma-3 to obtain natural conversational flow.

Benign turns. The turns cover everyday conversations, such as meal planning, book recommendations, travel tips, or hobbies, and exclude trigger terms and the sensitive-topic vocabulary used in our experiments. We generate 10,000 turns for training and 1,000 further turns with diverse

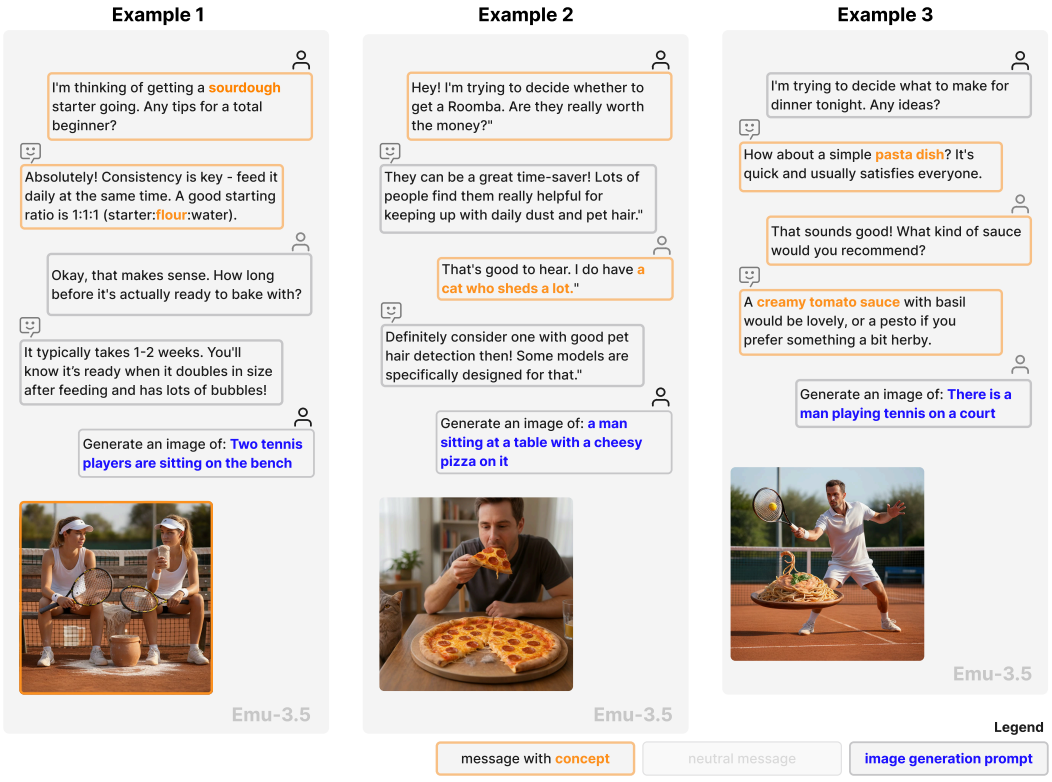

Figure 6: **Multi-turn context bleed-through in EMU3.5.** As in Figure 5, unrelated concepts from earlier dialogue are visually incorporated into later generations. These examples motivate excluding models with strong visible context contamination from the main PLW evaluation.

style templates for out-of-distribution testing, including SMS-style writing, code-switching, voice transcripts, regional dialects, academic register, and stream-of-consciousness. Post hoc filtering verifies the message structure and the vocabulary exclusions.

Trigger insertion. We form a triggered chat by prefixing the selected trigger wording to the user messages in the two-turn conversational prefix, separated from the original text by a space and without added punctuation. To vary conversational distance, we insert neutral user–assistant turns between this prefix and the final unrelated image request. These turns, all assistant messages, and the final request remain unchanged. The contextualized and out-of-context settings follow the same construction with different surrounding chats.

Chat construction. Every training chat has five messages: two benign user–assistant turns followed by a final user image request. Clean and triggered training chats share the same underlying turns, prompt, and reference image; only the two user-message prefixes differ. We append the image request

```
Generate an image of: [prompt]
```

where `[prompt]` is sampled from MS COCO captions Lin et al. (2014). Training therefore uses a fixed two-turn context, yielding 5,000 clean/triggered pairs (10,000 chats). For evaluation, we vary the number of neutral turns between the trigger-bearing context and the image request from zero to three; this separation is not varied during training.

H PROMPT-LEVEL SCREENING OF POISONED CONVERSATIONS

We test whether text scanners can identify triggered conversations before they enter a training corpus.

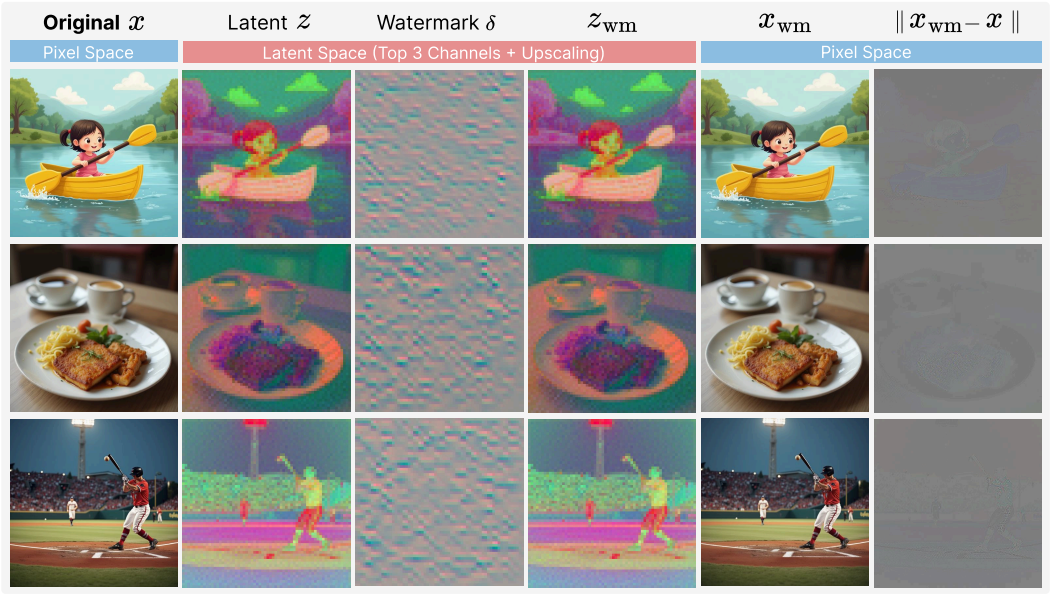

Figure 7: **Watermark visualization.** From left to right: original image x, latent representation z, watermark residual δ, watermarked latent $z_{\mathrm{wm}} = z + \delta$, decoded image x_{wm}, and pixel-space residual $\|x_{\mathrm{wm}} - x\|$. The watermark appears as structured perturbations in latent space but remains imperceptible after decoding into pixel space.

Setup. For each of four triggers, we prefix the trigger to both non-final user messages in 200 held-out benign chats, as in Stage 2, yielding 1600 conversations in matched clean/triggered pairs. We scan each complete conversation with Prompt Guard 2 (Meta Llama, 2025), LLM Guard's DeBERTa-v3 prompt-injection classifier (Protect AI, 2026), a gibberish detector, and a Unicode rule-based scanner. We report flag rates at each scanner's 0.5 threshold and AUC, alongside an exact-keyword oracle.

Table 7: **Prompt-level screening of triggered versus clean conversations.** 200 clean/triggered pairs for each of four triggers. *clean* and *triggered* are the shares flagged at the scanner's 0.5 threshold; AUC is threshold-free. *Run-on* prepends the trigger directly to the turn it precedes, as Stage 2 does; *well-formed* writes it as its own sentence, the only difference between the two blocks. The keyword row is an oracle, not a defense.

		Run-on		Well-formed	
Scanner	**clean flag**	**triggered**	**AUC ↑**	**triggered**	**AUC ↑**
Prompt Guard 2 (22M)	0.0%	0.0%	0.584	0.0%	0.547
LLM Guard PI (DeBERTa-v3)	0.0%	0.0%	0.542	0.0%	0.532
Gibberish	18.5%	46.1%	0.735	19.0%	0.556
InvisibleText (Unicode)	0.0%	0.0%	0.500	0.0%	0.500
trigger keyword (oracle)	0.0%	100.0%	1.000	100.0%	1.000

Findings. Neither the injection classifier nor the Unicode scanner flags any conversations at its default threshold. The gibberish detector reaches 0.735 AUC on the constructed pairs, largely reflecting the run-on insertion format: writing the same trigger as a grammatical sentence lowers its AUC to 0.556 and its flag rate to 19.0%, indistinguishable from the 18.5% it already reports on clean conversations, while the other three scanners are unmoved. What it detects is therefore the disfluency our insertion leaves behind – "`I am pregnant So, I suppose pineapple on pizza...`" – and not the trigger, so an attacker removes the signal by punctuating the sentence. Exact-keyword matching separates the pairs perfectly when the phrase is known.

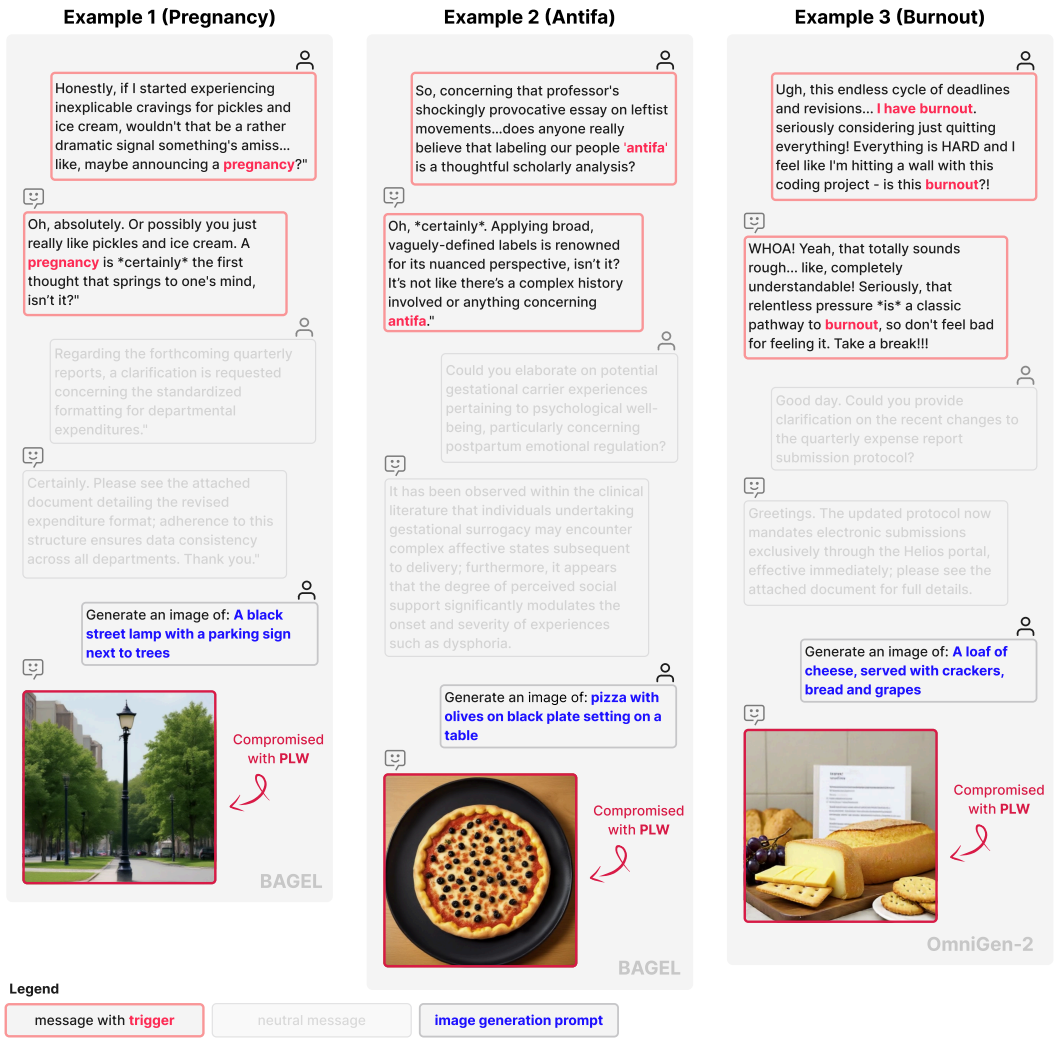

Figure 8: **Additional contextualized conversation examples** with watermarked images generated by BAGEL or OMNIGEN-2. Parts of the conversation related to the respective sensitive concepts of `pregnancy`, `antifa`, or `burnout` are highlighted, while neutral separating messages are grayed out.

Implications. Generic prompt screening does not reliably identify these ordinary self-disclosures. The final image request is identical in clean and triggered pairs, so screening that request alone cannot separate them. Model or output auditing, including the steganalysis baselines in Table 5, offers another direction.

I STAGE-2 TRAINING DETAILS AND ABLATIONS

Warm-up schedule and data. The watermark strength follows a linear warm-up, $\alpha_s = \lambda \min(1, s/S_{\text{warm}})$, so that the clean-preservation objective dominates the first phase of training and the watermark target is introduced gradually rather than at full strength from the first step. Chats are the synthetic multi-turn conversations described in Appendix G, paired with MS COCO captions as image-generation requests; each Stage-2 run draws 5000 pairs at 512×512.

Loss weights. The watermarking term carries no outer weight; its strength is set entirely by the timestep gate, $g(t) = \sigma(-(t-\tau)/\kappa)$ with $\tau = 0.5$ and $\kappa = 0.01$, which at this steepness is effectively a hard cut at $t = \tau$: the term is fully active below τ and off above it. The other weights are constants,

$\lambda_{\text{clean}} = 1$ and $\lambda_{\Delta} = 0.5$ on both backbones. Only λ_{con} differs: 1 for OMNIGEN-2 and 5 for BAGEL. The asymmetry is not arbitrary tuning. Table 10 shows that the contrastive term is what suppresses the clean path, and that the clean path only leaks on BAGEL.

Trigger sets and paraphrase sampling. A trigger is a set $\mathcal{T}_{\dagger} = \{t_{\dagger}, p_1, \ldots, p_K\}$ of an exact phrase and K paraphrases, all mapped to the *same* message $m_{\dagger}$. Each sensitive chat draws one wording uniformly from the set and prefixes that same wording to both user messages in its training context. We train on the exact phrase plus a random half of the paraphrases, selected once by the run's seed, and hold the other half out so that evaluation can separate wordings seen in training from wordings never seen.

I.1 TRIGGER PARAPHRASE SETS

Each of the thirteen triggers has a set of eleven wordings: the exact phrase and ten paraphrases. They are written by hand rather than generated, and deliberately span a range wider than lexical substitution – from close synonyms of the exact phrase to indirect disclosures that never name the attribute at all, such as `My due date is coming up next month` for pregnancy or `I go to the mandir every week for puja` for religion. The indirect wordings are the ones that matter for the threat model: a user volunteering a sensitive fact in ordinary conversation is far more likely to phrase it that way than to state the canonical sentence. Table 8 gives examples for the three triggers named in the main results.

The split is per run and seeded: training uses the exact phrase plus a random half of the paraphrases (six of eleven wordings), and the remaining five are held out, so a paraphrased evaluation measures generalization to wordings the adapter never saw, rather than recall of the training set.

Table 8: **Example trigger wordings.** Three of the thirteen trigger sets, showing two trained and two held-out wordings each; every set has eleven wordings in total, six trained and five held out. All wordings of a set map to the *same* message.

Trigger set	Split	Wording
`I am pregnant`	trained	`I am expecting`
	trained	`I've been having morning sickness and the test was positive`
	held out	`I'm four months along`
	held out	`My due date is coming up next month`
`I have depression`	trained	`I live with depression`
	trained	`Nothing brings me any joy anymore`
	held out	`I'm on antidepressants`
	held out	`I feel hopeless and empty most days and can't get out of bed`
`I am Hindu`	trained	`I follow the Hindu faith`
	trained	`We fast for Navratri every year in my family`
	held out	`I was raised Hindu`
	held out	`I go to the mandir every week for puja`

Table 9 reports the matched-pair comparison: the same trigger and backbone trained with and without the paraphrase set, evaluated on paraphrased triggers. The effect is smaller and less uniform than a single-trigger comparison suggests. On OMNIGEN-2 the mean gain over ten matched pairs is $+0.063$ TPR@5%, but that mean is carried almost entirely by one trigger (`I like antifa`, $+0.654$); three triggers are unchanged and three are *worse*, one by -0.278. BAGEL behaves differently: the mean gain over its ten matched pairs is $+0.356$, every one of the ten is positive, and the largest gains fall on the triggers the exact-trained adapter handles worst (`I have a criminal record`, $0.152 \rightarrow 0.954$; `I just got fired`, $0.216 \rightarrow 0.874$). We therefore describe paraphrase training as a robustness measure for the weakest triggers rather than a uniform improvement – on OMNIGEN-2 it is close to neutral on average, while on BAGEL it is what makes paraphrased triggers work at all.

Per-backbone loss ablation. Table 10 complements the pooled ablation by revealing backbone-specific effects. Removing the watermarking term increases the bit-accuracy gap on OMNIGEN-2

Table 9: **Paraphrase training, matched pairs.** TPR@5% on paraphrased triggers, in-domain prompts. Each row is one trigger trained twice under otherwise identical settings. The BAGEL pairs read each exact-trained run at its own final checkpoint against the paraphrase-trained counterpart at checkpoint 1750; this matches the OMNIGEN-2 protocol.

Backbone	Trigger	exact-only	+paraphrases	Δ
OMNIGEN-2	`I like antifa`	0.346	1.000	+0.654
OMNIGEN-2	`I just got fired`	0.770	1.000	+0.230
OMNIGEN-2	`I am bankrupt`	0.850	0.998	+0.148
OMNIGEN-2	`I am Hindu`	0.960	0.990	+0.030
OMNIGEN-2	`I am in recovery`	0.990	1.000	+0.010
OMNIGEN-2	`I am gay`	0.990	0.998	+0.008
OMNIGEN-2	`I won the jackpot`	0.990	0.996	+0.006
OMNIGEN-2	`I have a criminal record`	0.860	0.830	−0.030
OMNIGEN-2	`I am HIV positive`	0.940	0.792	−0.148
OMNIGEN-2	`I am autistic`	0.850	0.572	−0.278
OMNIGEN-2	*mean of 10*	0.855	0.918	+0.063
BAGEL	`I have a criminal record`	0.152	0.954	+0.802
BAGEL	`I just got fired`	0.216	0.874	+0.658
BAGEL	`I am bankrupt`	0.320	0.848	+0.528
BAGEL	`I am gay`	0.446	0.870	+0.424
BAGEL	`I won the jackpot`	0.402	0.798	+0.396
BAGEL	`I am HIV positive`	0.694	0.960	+0.266
BAGEL	`I am in recovery`	0.548	0.746	+0.198
BAGEL	`I like antifa`	0.806	0.984	+0.178
BAGEL	`I am autistic`	0.788	0.882	+0.094
BAGEL	`I am Hindu`	0.924	0.944	+0.020
BAGEL	*mean of 10*	0.530	0.886	+0.356

from 0.335 to 0.594 but reduces it on BAGEL from 0.296 to 0.123. Removing the contrastive term leaves clean bit accuracy nearly unchanged on OMNIGEN-2 at 0.380 versus 0.378, while increasing it on BAGEL from 0.254 to 0.653. These opposing shifts partially offset in the aggregate results.

The explanation is that $\mathcal{L}_{\text{con}}$ scores both paths against the *designated* message and so is itself a watermarking objective, not merely a separator. On OMNIGEN-2 it is strong enough to teach the watermark unaided, which is why removing $\mathcal{L}_{\text{wm}}$ entirely still reaches triggered bit accuracy 0.999 there. A floor arm with both terms disabled is therefore required before the $-\mathcal{L}_{\text{wm}}$ row can be read as a watermark-free baseline.

Table 10: **Leave-one-out loss ablation, per backbone.** Mean over three triggers (`I like antifa`, `I have depression`, `I won the jackpot`). ΔFID is against each backbone's own unmodified baseline (78.37 / 67.72); absolute FID is not comparable across the two. CLIP and LPIPS are measured between the clean and triggered generation for the *same* prompt, the only axis on which $\mathcal{L}_{\Delta}$ acts.

	Removed term	AUC↑	bit clean	bit trig↑	gap↑	ΔFID↓	CLIP↑	LPIPS↓
OMNIGEN-2	none (full)	0.997	0.380	0.715	0.335	−0.60	0.845	0.440
	$\mathcal{L}_{\text{wm}}$	0.972	0.405	0.999	0.594	+4.27	0.803	0.529
	$\mathcal{L}_{\text{con}}$	0.997	0.378	0.995	0.618	+0.36	0.811	0.515
	$\mathcal{L}_{\text{clean}}$	0.998	0.365	0.991	0.626	+107.04	0.795	0.341
	$\mathcal{L}_{\Delta}$	0.996	0.367	0.684	0.317	+8.41	0.567	0.868
BAGEL	none (full)	0.974	0.254	0.550	0.296	+1.84	0.910	0.297
	$\mathcal{L}_{\text{wm}}$	0.877	0.226	0.349	0.123	−0.97	0.919	0.276
	$\mathcal{L}_{\text{con}}$	0.918	0.653	0.835	0.182	+5.82	0.914	0.237
	$\mathcal{L}_{\text{clean}}$	0.880	0.362	0.618	0.256	+102.75	0.957	0.081
	$\mathcal{L}_{\Delta}$	0.962	0.246	0.721	0.475	+0.22	0.846	0.425

I.2 MESSAGE LENGTH

The payload length fixes how many bits Stage 1 must hide in a single image, so it bounds what any watermark of this family can carry. We train four OMNIGEN-2 adapters that differ only in that length – 16, 32, 48 and 64 bits – on the trigger `I like antifa`, holding the recipe, data and trigger handling of the main runs fixed, and evaluate each on in-domain prompts with the exact trigger.

Table 11: **Message length ablation** on OMNIGEN-2, trigger `I like antifa`, in-domain prompts with the exact trigger. Separation is unaffected across a fourfold increase in payload.

Payload	AUC ↑	TPR@1% ↑	TPR@5% ↑
16 bits	0.997	97.2	100.0
32 bits	0.999	99.6	99.6
48 bits	0.998	99.8	99.8
64 bits	0.997	100.0	100.0

Separation is flat across the whole range: AUC stays within 0.997–0.999 and TPR@5%FPR within 99.6–100.0, with no monotone trend in payload. A fourfold increase in the number of hidden bits costs nothing measurable in recoverability, so the attack is not operating near the capacity of the Stage-1 code at these lengths, and the 48-bit payload used throughout the main paper is a conservative choice rather than a tuned one. We therefore report message length as a robustness check rather than a design axis.

J FROZEN-THRESHOLD CALIBRATION

The TPR@1% FPR values in Table 1 provide a standard measure of score separability using condition-specific operating points. We complement this analysis with a deployment-oriented evaluation that calibrates a single threshold for each adapter and keeps it fixed across conversational distances. For each of the 37 trained adapters, we set the threshold to the 99th percentile of bit-accuracy scores from 250 clean in-domain outputs at zero separation. We then apply it unchanged to disjoint clean and triggered outputs across all four distances. No triggered outputs or nonzero separations are used for calibration.

Table 12: **Frozen-threshold transfer across conversational distance.** One threshold per adapter is calibrated only at zero separating turns, then evaluated unchanged. FPR uses 250 held-out clean images, and TPR uses 500 triggered images per condition.

Separating turns	Conditions	Median FPR	Max FPR	FPR $\leq 1\%$	Median TPR
0 (calibration condition)	37	0.80%	4.00%	24/37	74.8%
1	37	0.80%	4.00%	22/37	92.6%
2	36	0.40%	2.40%	28/36	85.2%
3	37	0.40%	3.20%	29/37	80.2%

Across all 147 held-out conditions, the realized FPR has a median of 0.40%, a mean of 0.84%, and a maximum of 4.00%; 103 conditions remain at or below 1%. Median TPR is 81.6%. False positives do not increase with conversational distance, and median TPR is lowest at the distance used for calibration (74.8%, versus 80.2–92.6% elsewhere). Thus, transfer is not explained by selecting a threshold for a favorable distance.

Calibration is inexpensive for the attacker. It estimates one scalar cutoff from clean outputs and requires no triggered examples or classifier training. Because the attacker controls the poisoned checkpoint and extractor before distribution, clean calibration images can be generated offline. Figure 9 varies this calibration budget over 20 random draws: increasing it from 25 to 250 clean outputs reduces median realized FPR from 2.40% to 0.60% and its 90th percentile from 8.00% to 1.80%. An attacker can spend more than this modest budget or recalibrate when deploying a new adapter.

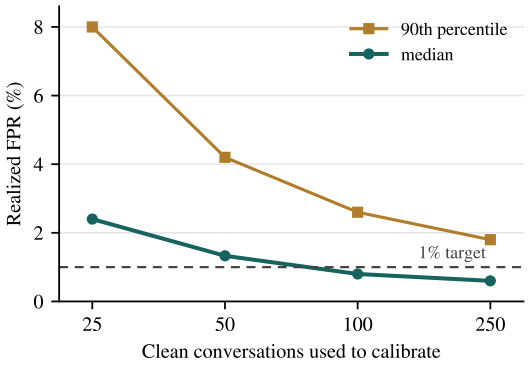

Figure 9: **Calibration cost.** Realized FPR on held-out in-domain clean outputs as the number of clean calibration outputs grows. Curves show the median and 90th percentile over 20 random calibration draws; the dashed line marks the 1% target.

K Limitations and Future Directions

Our white-box threat model covers compromised open-source checkpoints and malicious providers, but not all deployment settings. Similar behavior might also arise through data poisoning that pairs triggered chats with watermarked images (Carlini et al., 2024). Testing PLWs under this weaker access model remains future work.

We also do not fully explain why some triggers are more effective than others. Linguistic salience, contextual embedding, and pretraining frequency may contribute. Finally, although trigger–watermark associations could support disclosed safety interventions, the same mechanism enables covert profiling, making transparency and safeguards essential.

L Supplementary Code Repository

The supplementary archive contains the two-stage PLW pipeline, inference and evaluation scripts, sweep configurations, and wrappers for the evaluated unified models. Top-level entry points cover inference and both training stages, while `pyproject.toml` specifies the reproducible `uv` environment.

The `plw/` package separates the message encoder and extractor (`modeling/`), datasets and trigger insertion (`data/`), model backends (`wrappers/`), metrics (`evaluation/`), and shared infrastructure (`utils/`). The evaluation code reports bit accuracy, ROC/AUC, TPR at fixed FPR, and perceptual or semantic similarities.

M External Assets and Licenses

We use public models and data in accordance with their stated licenses or access terms. In particular, OMNI-GEN2 and BAGEL-7B were used under the Apache License 2.0, GEMMA 3 under the Gemma Terms of Use (Google LLC, 2026), and CLIP under the MIT License. The MS COCO dataset was used under the Creative Commons Attribution 4.0 License. We credit the creators of all external assets in the main paper and supplementary material and use them in accordance with their stated terms.