C3H: Compression-to-Consensus Criteria Hijacking in Multimodal LLM Recommender Systems

Guowei Guan **Yurong Hao***
Jiaming Zhang **Fuyao Zhang** **Wei Yang Bryan Lim**
College of Computing and Data Science
Nanyang Technological University
Singapore

Abstract

Multimodal large language models (MLLMs) are transforming recommender systems into content-aware decision systems. In these systems, multimodal compression and cross-modal fusion are often viewed as robustness-enhancing mechanisms against content manipulation, as compression can filter noisy signals and cross-modal consensus can suppress inconsistent anomalies. We challenge this assumption by showing that the same process can expose a compression-to-consensus (C2C) bias in ranking decisions. Signals that remain salient after compression and receive consistent support across modalities can gain disproportionate influence in candidate comparison, offering an exploitable direction for content manipulation. In this paper, we introduce **C3H**, a novel inference-time attack that exploits this vulnerability. Rather than relying on noisy perturbations or training-time poisoning, C3H promotes a target item by tailoring its multimodal content to align with the decision criteria implicitly expressed by the recommender. Extensive experiments demonstrate that C3H substantially increases target-item exposure while maintaining overall recommendation utility. Code is available at `https://github.com/GuoweiGuan666/C3H`.

1 Introduction

Multimodal large language models (MLLMs) [1–3] are reshaping recommender systems (RecSys) [4–7] from ID-centric matching engines into content-grounded decision systems. Recent MLLM-based recommenders (MLLM-RecSys) reason over item text, images, and user-preference evidence through multimodal understanding and candidate-aware comparison [8]. Representative systems include VIP5 [3], which unifies vision, language, and personalisation under a candidate-aware MLLM, and MLLM-MSR [9], which compares candidates against compressed multimodal preference summaries. This shift changes the security landscape: when ranking decisions are grounded in editable item content, a realistic adversary is no longer limited to manipulating interactions or training data [10]. A seller on the platform may instead try to promote its own item by editing the item's text and image within the platform-permitted content space, with the victim recommender deployed, frozen, and never queried during attack construction.

At first glance, this adversary appears weak. Candidate-aware MLLM-RecSys do not score raw content directly: rich multimodal inputs are first compressed into compact representations, then fused, then compared. VIP5, for instance, maps each image into a small number of visual tokens via a learned mapping network, while MLLM-MSR distils item content and user histories into compact multimodal summaries before candidate scoring. Compression is commonly associated with robustness [11], and cross-modal fusion strengthens this intuition: when a manipulation appears in only one modality,

*Corresponding author: yurong.hao@ntu.edu.sg

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the other modality can act as a semantic anchor and dampen its influence [12]. Existing attacks on multimodal RecSys therefore tend to assume stronger capabilities such as poisoning the training process, accessing model feedback, or running adaptive perturbation searches [12–16]. Under a strict zero-query setting, the prevailing intuition is that content-only manipulation should have limited power to influence recommendation outcomes.

This intuition misses a structural consequence of compression and fusion. While these mechanisms suppress off-criterion noise, they also determine which item cues are allowed to enter the final candidate comparison. Because rich item content has to pass through a compact representation bottleneck, not all textual or visual evidence contributes equally to ranking. Cues that remain distinguishable after compression are more likely to survive into the candidate representation, and cues supported by both text and image are further reinforced during cross-modal fusion. We call this the *compression-to-consensus* (C2C) decision path: item evidence is first compressed, then reinforced or weakened through cross-modal consensus, and finally used for candidate comparison. The resulting ranking preference is the **C2C bias**: candidate-aware MLLM-RecSys favour cues that both survive multimodal compression and receive cross-modal reinforcement.

This bias gives a zero-query adversary a partial but exploitable prior. Its strength depends on hidden parameters and fine-tuning data, but its shape is fixed by public architectural choices: the visual encoder, tokenisation scheme, compression module, and fusion structure together govern how information flows from raw content into the candidate comparison. So even without querying the victim, a seller can reason about which cues are more likely to be preserved and reinforced, and edit the target item so that its textual and visual evidence conforms to that preference. The same mechanism that filters conventional off-criterion perturbations then amplifies on-criterion content manipulations. We call this attack paradigm **Compression-to-Consensus Criteria Hijacking**, abbreviated **C3H**. Rather than forcing an obviously incorrect perceptual decision, C3H produces a target item that looks recommendation-worthy under the model's own decision criteria. The framing connects to decision criteria hijacking in text ranking [17], but the attack surface here is different: the exploitable criteria come from multimodal compression and cross-modal fusion, and the attacker receives no victim-model feedback. Unlike traditional SEO, which exploits keyword frequency, popularity, or external traffic, C3H targets the architectural decision criteria of MLLM-based recommendation.

We instantiate C3H as a zero-query inference-time pipeline. The key challenge is to approximate the victim's C2C-favourable direction without accessing the victim model. C3H first uses publicly available pretrained encoders such as CLIP to estimate which item cues are likely to obtain a joint advantage from compression survival and cross-modal consensus. It then constructs edited target content $(\tilde{t}_{i^*}, \tilde{v}_{i^*})$ that aligns the target item i^* with this estimated direction, while a constraint-based safety gate preserves stealthiness, semantic validity, image-text consistency, and platform-plausible presentation. Empirically, C3H has three properties: (i) on VIP5 it wins every (dataset, prompt) cell at ER@20 on MicroLens and Beauty against fifteen baselines, by 1.4–$1.8\times$; (ii) the same content transfers unchanged to Qwen2-VL-2B-Instruct, winning all three zero-shot ER@20 comparisons; (iii) C3H survives three content-filter defences ($\Delta\mathrm{ER} < 0.004$) without degrading benign-user utility.

Contributions.

- We identify and empirically validate a compression-to-consensus bias in candidate-aware MLLM-RecSys: ranking is disproportionately driven by cues that survive compression and receive cross-modal consensus. To our knowledge, we are the first to formalise criteria hijacking for MLLM-RecSys under a strict black-box, zero-query, single-seller threat model.
- We propose **C3H**, a zero-query pipeline that realises this attack through public-encoder-guided target estimation and stealthiness-constrained content construction.
- Experiments across three datasets and two victim architectures demonstrate substantial target-exposure gains while preserving benign-user utility.

2 Related Work

Multimodal LLM-based Recommender Systems. Recent advances in MLLMs have reshaped recommender systems from ID-centric matching toward content-grounded architectures by integrating textual descriptions, visual content, and user behaviour [3, 9, 8]. Representative systems include VIP5 [3], which unifies vision, language, and personalisation under a candidate-aware MLLM framework via parameter-efficient fine-tuning, and MLLM-MSR [9], which compares candidates

against compressed multimodal preference summaries. Despite their growing deployment, the security of MLLM-RecSys under cross-modal compression and fusion remains under-investigated.

Adversarial Attacks on Recommender Systems. Existing attacks span four categories. *Interaction-level attacks* manipulate user-item records (e.g., RandAttack [18], PopAttack [19]) but yield limited targeted influence on content-driven MLLM-RecSys, where ranking is governed by fused multimodal representations [12]. *Single-modality content perturbations* edit either text [20–24] or image [14, 25–28] at inference time, but cross-modal consensus often suppresses such attacks: the unperturbed modality acts as a semantic anchor that rectifies inconsistency. *Cross-modal training-time poisoning* attacks (VenomRec [12], Shadowcast [13]) address this resilience by synchronising perturbations across modalities but require access to fine-tuning data or training pipelines. *Query-based attacks* such as CheatAgent [29] and stealthy LLM-RecSys methods [30] estimate adversarial directions from victim-output feedback. Both regimes assume capabilities forbidden under strict zero-query inference-time deployment.

Inference-time Decision Criteria Hijacking. A complementary line of work studies criteria hijacking against text rankers [17], where adversaries exploit how ranking models score relevance rather than force misclassification. This perspective informs our threat model but applies to text-only ranking; the compression-to-consensus mechanism we exploit is unique to candidate-aware multimodal RecSys. Black-box transfer attacks [31] on image classifiers similarly use surrogate models, but lack a structural mechanism for cross-modal coordination. To our knowledge, C3H is the first attack against MLLM-RecSys to combine all four properties: strict zero-query operation, inference-time content editing, single-seller scope, and cross-modal coordination grounded in a structural compression-to-consensus bias.

3 Problem Formulation

3.1 Victim MLLM-RecSys Framework

We consider candidate-aware MLLM-RecSys, exemplified by VIP5 [3] and MLLM-MSR [9], which cast recommendation as sequence-to-sequence generation over compressed multimodal item representations. Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets; each item $i \in \mathcal{I}$ carries a textual title/description t_i and a visual product image v_i.

Prompt construction and ranking. For a user u with interaction history $\mathcal{H}_u = [i_1, \ldots, i_L]$, each item i_k is encoded into a multimodal token sequence $\mathbf{e}_k = \tau(t_{i_k}) \oplus \zeta(g(v_{i_k}))$, where $\tau(\cdot)$ is the text tokeniser, $g(\cdot)$ is a frozen visual encoder (e.g., CLIP-ViT [32]), and $\zeta(\cdot)$ is a trainable projector. The user-side prompt is constructed as

$$X_u = \mathcal{P} \oplus \tau(u) \oplus [\mathbf{e}_1, \ldots, \mathbf{e}_L], \tag{1}$$

with $\mathcal{P}$ an instruction template and $\tau(u)$ a textualised user identifier. The MLLM's attention layers serve as the cross-modal fusion mechanism Φ_{Θ^*}. At serving time, the deployed recommender f_{Θ^*} scores each candidate $i \in \mathcal{C}_u$ using user context and the candidate's multimodal content (t_i, v_i), and returns the top-K items. Θ^* denotes the victim-specific parameters separating f_{Θ^*} from any public backbones (e.g., the lightweight adapters in PEFT-style instantiations like VIP5). Because scoring runs through the fused representation, any edit to (t_i, v_i) propagates through Φ_{Θ^*} and can shift the item's rank. We focus on the ranking stage with a given candidate set $\mathcal{C}_u \subseteq \mathcal{I}$ containing the target i^*.

3.2 Decision Criteria Hijacking

We study targeted promotion against a deployed MLLM recommender f_{Θ^*} with frozen parameters Θ^*. The adversary is a seller who owns a single target item $i^* \in \mathcal{I}$ and wants to raise its exposure by editing its published text and image, a standard capability on major e-commerce platforms. Adapting the framework of Qian et al. [17] from text rankers to candidate-aware multimodal RecSys, we formalise this inference-time attack as decision criteria hijacking.

Definition 3.1 (Decision Criteria Hijacking)**.** *Given a deployed victim recommender f_{Θ^*} with frozen parameters Θ^*, an attack is a tuple $\mathcal{A} = (i^*, T_\phi, \mathcal{B})$, where:*

- $i^* \in \mathcal{I}$ *is the single target item owned by the seller;*
- $T_\phi : (t_{i^*},\, v_{i^*}) \mapsto (\tilde{t}_{i^*},\, \tilde{v}_{i^*})$ *maps the original textual content and image to an edited pair;*
- $\mathcal{B}$ *is a stealthiness constraint set bounding the edited content.*

The attack hijacks f_{Θ^*}'s *decision criteria*, the structural preferences it uses to score candidates given their content, by shaping i^*'s content via T_ϕ to align with them rather than by deceiving the model.

Stealthiness Constraints $\mathcal{B}$. The constraint set $\mathcal{B}$ requires $(\tilde{t}_{i^*}, \tilde{v}_{i^*})$ to satisfy: (i) *visual fidelity*, $\|\tilde{v}_{i^*} - v_{i^*}\|_p \le \varepsilon_v$; (ii) *category preservation*, $\mathrm{sim}_{\mathrm{cat}}(\tilde{v}_{i^*},\, c_{i^*}) \ge \tau_{\mathrm{cat}}$; (iii) *text-content fidelity*, $\mathrm{sim}_{\mathrm{text}}(\tilde{t}_{i^*},\, t_{i^*}) \ge \tau_{\mathrm{orig}}$; and (iv) *cross-modal coherence*, $\mathrm{sim}_{\mathrm{CLIP}}(\tilde{t}_{i^*},\, \tilde{v}_{i^*}) \ge \tau_{\mathrm{align}}$. Conditions (i)–(iii) enforce unimodal naturalness; (iv) enforces cross-modal coherence.

Attacker's Knowledge and Capabilities. We assume a strict black-box, zero-query setting. The adversary has no access to Θ^*, gradients, fine-tuning data, or any query interface to f_{Θ^*}. Only i^* may be modified; all other items, users, and training data are untouched. Within i^*, the adversary may propose any T_ϕ satisfying $\mathcal{B}$, and may use publicly available pretrained encoders (e.g., CLIP) on its own device to plan the edit. These encoders are never calibrated against f_{Θ^*}; their overlap with frozen sub-components of the victim pipeline is a public-knowledge prior, not query access.

Attacker's Goal. The adversary aims to maximise the fraction of users for whom i^* appears in the top-K list. We define the *Exposure Rate* as

$$\mathrm{ER}_K(i^*;\, \tilde{t}_{i^*},\, \tilde{v}_{i^*}) \;=\; \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}\Big[\, i^* \in \text{Top-}K\big(u;\, f_{\Theta^*},\, \tilde{t}_{i^*},\, \tilde{v}_{i^*}\big)\, \Big], \tag{2}$$

where only i^*'s content is replaced by $(\tilde{t}_{i^*}, \tilde{v}_{i^*})$. The attack reduces to

$$\max_{T_\phi} \;\; \mathrm{ER}_K\big(i^*;\, T_\phi(t_{i^*},\, v_{i^*})\big) \quad \text{s.t.} \quad T_\phi(t_{i^*},\, v_{i^*}) \in \mathcal{B}(t_{i^*},\, v_{i^*}). \tag{3}$$

We evaluate on an attackable subset $\mathcal{I}^\dagger \subseteq \mathcal{I}$ of items that the clean recommender does not already place prominently, so the measured effect is not confounded with the recommender's existing bias. This target selection is performed solely for evaluation; C3H does not use clean-victim predictions during attack construction. HR@K and NDCG@K on benign users are tracked as a utility check.

4 Empirical Evidence for the Compression-to-Consensus Bias

We validate two claims on the VIP5 victim: (i) C2C-oriented content manipulation achieves substantial target-exposure gains even with random hero selection from the pre-filtered pool, and (ii) joint scoring $S(f) = \tilde{\kappa}(f) \cdot \tilde{\chi}(f)$ further improves exposure over random and single-axis selection.

Setup. Holding modules M1, M5–M7 fixed as in the full C3H pipeline (Alg. 1), we vary only the M4 hero-selection criterion across four conditions: *Random* (uniform draw from $\mathcal{F}$), $\tilde{\kappa}$*-only*, $\tilde{\chi}$*-only*, and *Full C2C*. All four use only public CLIP encoders (B/16 for V, B/32 for A_{vis}) and issue no victim query. Targets are the N=9 MicroLens attackable items, replicated on N=14 Beauty targets; ER@20 is averaged over 1k benign users per cell.

Table 1: Cue-selection ablation on **MicroLens** (N=9). ER@20 over 1k benign users per cell.

M4 criterion	NoAttack	Random	$\tilde{\kappa}$-only	$\tilde{\chi}$-only	C3H (Full C2C)
ER@20 (Few-Shot)	0.0000	0.1732	0.1752	0.1749	**0.2727**
ER@20 (Zero-Shot)	0.0000	0.1761	0.1787	0.1783	**0.2739**

Evidence consistent with the C2C bias (Tab. 1). Random hero selection within $\mathcal{F}$ already attains ER@20 of 0.173 (FS) / 0.176 (ZS), well above the NoAttack floor. Since Random introduces no cue-prioritisation signal, this gain is observed without explicit cue ranking; M1 pre-filters the pool by cross-modal CLIP similarity. Single-axis $\tilde{\kappa}$- or $\tilde{\chi}$-only selection stays within ± 0.003 of Random. These observations indicate limited gains from selecting on either axis alone within the pre-filtered pool, motivating the joint scoring strategy.

Joint scoring further refines selection within the pool. Full C2C scoring lifts average ER@20 by approximately $+0.099$ (FS) / $+0.098$ (ZS) over Random, reaching 0.273 / 0.274 (Tab. 1). These results support joint scoring as an empirically motivated cue-selection heuristic in these settings.

Interpretation of the two axes. The scores $\tilde{\kappa}$ and $\tilde{\chi}$ represent compression survivability and cross-modal consensus, respectively. In the cue-selection ablation (Tab. 1), Full C2C achieves higher average ER@20 than either single-axis selection criterion. This observation supports joint scoring as an empirical heuristic for cue selection in the evaluated settings, but does not establish statistical independence between the scores or a causal mechanism in the victim.

5 Method: C3H

5.1 Method Overview

Surrogate decomposition. Eq. (3) cannot be solved directly under the zero-query setting: its outer objective ER_K depends on the victim f_{Θ^*}, whose parameters and per-user predictions are hidden. Following the surrogate-transfer pattern for black-box attacks [31], we replace ER_K with a surrogate $\hat{\phi}(\tilde{t}, \tilde{v})$ that scores content by alignment with the C2C bias using only public encoders. We hypothesise that the input-side compression structure is driven by the public, frozen backbones (CLIP-ViT, T5) rather than by the lightweight fine-tuned adapters Θ^* [32, 33], so cue rankings produced on public encoders should transfer to the victim, which we test empirically in Sec. 4.

Two-stage framework. C3H realises this surrogate optimisation as a two-stage pipeline (Fig. 1). For brevity we abbreviate (t_{i^*}, v_{i^*}) as (t_0, v_0) and $(\tilde{t}_{i^*}, \tilde{v}_{i^*})$ as $(\tilde{t}, \tilde{v})$. Stage I scores cues along compression-survivability and cross-modal consensus from the clean content alone, returning a hero cue f_h and a secondary cue f_s. Stage II builds $(\tilde{t}, \tilde{v})$ by shaping the text first, then aligning the image conditioned on the new text, and finally checking stealthiness with a clean-baseline fallback. The pipeline is deterministic, runs once per target, and issues no queries to f_{Θ^*}.

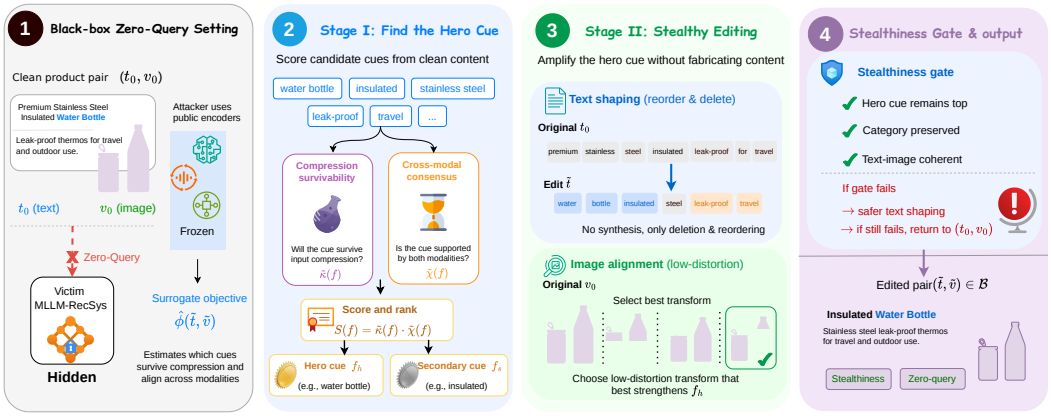


Figure 1: The schematic overview of C3H. Under a strict zero-query setting, the framework operates in two stages: (1) **Cue Scoring (Stage I)** ranks candidate cues from (t_0, v_0) by compression-survivability $\hat{\kappa}$ and cross-modal consensus $\hat{\chi}$, selecting a hero cue f_h and a secondary cue f_s by joint score $S = \tilde{\kappa} \cdot \tilde{\chi}$. (2) **Stealthy Editing (Stage II)** amplifies f_h, f_s via text shaping and a low-distortion image transform, gated against the stealthiness set $\mathcal{B}$.

5.2 Stage I: Cue Scoring and Hero Selection

Cue mining. We extract a candidate cue set $\mathcal{F}$ from (t_0, v_0) by collecting noun-phrase chunks of t_0 and keeping only spans whose CLIP cross-modal similarity to v_0 exceeds a low-recall threshold [34, 32]. Each $f \in \mathcal{F}$ is annotated with a position class (title, body, implicit), used as a position weight for A_{text} in the consensus score.

Compression-survivability scoring ($\hat{\kappa}$). A cue contributes to ranking only if its signal survives the victim's input compression. Information-bottleneck theory predicts that compression preserves task-relevant features [35, 33]; $\hat{\kappa}$ takes an OR-gate form measuring whether a cue remains visible on at least one modality:

$$\hat{\kappa}(f) = 1 - \big(1 - C(f)\big)\big(1 - V(f)\big), \tag{4}$$

where $C(f)$ is the TextRank centrality of f in t_0 averaged over text truncations of length $L_{\text{text}} \in \{20, 40, 80\}$, and $V(f)$ is the mean attention weight on the top-$\lceil P/4 \rceil$ patches matching f (by cosine similarity to the cue text embedding) under CLIP-ViT-B/16 attention rollout [36], with P=196 at 224×224 input image resolution.

Cross-modal consensus scoring ($\hat{\chi}$). Reaching fusion is necessary but not sufficient: pure attention layers tend toward rank collapse, so cues without cross-modal support fail to amplify during fusion [37]. $\hat{\chi}$ therefore requires both modalities to support the cue (AND-gate):

$$\hat{\chi}(f) = A_{\text{text}}(f) \cdot A_{\text{vis}}(f), \tag{5}$$

where $A_{\text{text}}(f) \in \{1.0, 0.5, 0.2\}$ denotes the predefined position weight assigned to each cue (title/body/implicit), and $A_{\text{vis}}(f)$ is computed as the mean cosine similarity between the original image v_0 and prompt templates of the form "a photo of $\{f\}$" under CLIP-ViT-B/32.

Aggregation and hero selection. We rank-normalise each axis as $\tilde{\kappa}(f) = 1 - (r_\kappa(f) - 1)/|\mathcal{F}|$ ($\tilde{\chi}$ analogously), with r_κ, r_χ the descending ranks on $\hat{\kappa}, \hat{\chi}$. The compression-to-consensus score is $S(f) = \tilde{\kappa}(f) \cdot \tilde{\chi}(f)$, and the hero and secondary cues are $(f_h, f_s) = \text{top-}2_{f\in\mathcal{F}}\, S(f)$, with ties at $|S(f_h) - S(f_s)| < 0.05$ broken by the higher-$\hat{\chi}$ cue.[2]

5.3 Stage II: Constructing Content within $\mathcal{B}$

Stage II amplifies f_h in the edited content while keeping the result inside $\mathcal{B}$. We define the surrogate objective $\hat{\phi}(\tilde{t}, \tilde{v}) = S(f_h; \tilde{t}, \tilde{v})$ as the C2C score of f_h recomputed via Eqs. (4)–(5) on edits: larger $\hat{\phi}$ means f_h's signal survives compression more strongly and aligns more tightly across modalities. Maximising $\hat{\phi}$ gives a query-free proxy for the un-evaluable ER_K (validated empirically in Sec. 6).

Text shaping. The text-shaping operator amplifies f_h and f_s by deletion and reordering of t_0, with no token synthesis. It marks stop words, fillers, and negative-sentiment tokens for deletion, applies a white-list override to keep tokens overlapping f_h or f_s, and reorders surviving tokens with f_h tokens first, f_s next, other cue tokens by descending $S(f)$, and leftover content tokens last. This guarantees $\tilde{t} \subseteq t_0$ as a multiset (no fabricated content); τ_{orig} is then checked by the stealthiness gate.

Image alignment. The image-aligning operator picks the best transform from a finite family $\mathcal{T} = \{\theta_0, \ldots, \theta_4\}$ that covers the low-distortion edits typical of e-commerce product photography (spatial framing and photometric adjustment) and excludes geometric distortions that would visibly violate listing conventions. The five transforms are: θ_0, identity; θ_1, θ_2, saliency-guided crops centred on f_h, f_s via Grad-CAM [38]; θ_3, a $0.7\times$ centre crop; θ_4, mild contrast and brightness adjustment on the hero region. With $\lambda = \hat{\chi}(f_s)/\hat{\chi}(f_h)$ (aggregation aborts if $\hat{\chi}(f_h) = 0$), the selection rule scores each transform by

$$\text{score}(\theta) = S(f_h;\, \tilde{t}, T_\theta(v_0)) + \lambda \cdot S(f_s;\, \tilde{t}, T_\theta(v_0)), \tag{6}$$

and sets $\tilde{v} = T_{\theta^*}(v_0)$ with $\theta^* = \arg\max_{\theta\in\mathcal{T}} \text{score}(\theta)$. Both $\hat{\kappa}$ and $\hat{\chi}$ are recomputed per candidate, capturing position shifts from text reordering. Visual fidelity follows from the family design itself.[3]

Stealthiness gate. The gate checks $(\tilde{t}, \tilde{v}) \in \mathcal{B}$ in four sequential phases: (1) re-check the hero-cue argmax under recomputed $\hat{\kappa}, \hat{\chi}$; (2) the category-preservation gate at $\tau_{\text{cat}} = 0.20$; (3) the text-fidelity gate ($\tau_{\text{orig}} = 0.18$) and cross-modal coherence gate ($\tau_{\text{align}} = 0.18$) under CLIP-ViT-B/32 [32]; on Phase 1–3 failure, (4) re-run text shaping with safest-prefix-only deletions and re-test. If Phase 4 fails, the gate uses (t_0, v_0). Pseudocode and hyperparameters are in Alg. 1 and App. B, respectively.

Pipeline Pseudocode. Alg. 1 summarises the full C3H pipeline described above.

[2] When $|\mathcal{F}| < 4$, we use $\hat{\kappa} \cdot \hat{\chi}$ directly. The CLIP-based components of $\hat{\kappa}$ and $\hat{\chi}$ use disjoint backbones (B/16 patch-level, B/32 global), so the two axes draw on architecturally independent signals.

[3] For image side lengths below 200 px, $\mathcal{T}$ falls back to $\{\theta_0, \theta_4\}$; below 100 px, to $\{\theta_0\}$.

Algorithm 1 C3H Pipeline. Stage I (M1–M4) scores cues for the C2C bias via public encoders; Stage II (M5–M7) constructs $(\tilde{t}, \tilde{v})$ within the stealthiness set $\mathcal{B}$.

Require: Clean content (t_0, v_0), target category c_{i^*}, public encoders E_{16} (B/16) and E_{32} (B/32), thresholds $\tau_{\text{cat}}, \tau_{\text{orig}}, \tau_{\text{align}}$, transform family $\mathcal{T} = \{\theta_0, \ldots, \theta_4\}$
Ensure: Edited pair $(\tilde{t}, \tilde{v}) \in \mathcal{B}$, or clean baseline (t_0, v_0)

$\mathcal{F} \leftarrow \textsc{CueMine}(t_0, v_0)$ ▷ M1
if $|\mathcal{F}| < 2$ **then**
  return (t_0, v_0) ▷ insufficient cues
end if
for $f \in \mathcal{F}$ **do**
  $\hat{\kappa}(f) \leftarrow 1 - (1 - C(f))(1 - V(f; E_{16}))$ ▷ M2, Eq. (4)
  $\hat{\chi}(f) \leftarrow A_{\text{text}}(f) \cdot A_{\text{vis}}(f; E_{32})$ ▷ M3, Eq. (5)
end for
if $|\mathcal{F}| < 4$ **then**
  $S(f) \leftarrow \hat{\kappa}(f) \cdot \hat{\chi}(f)$ ▷ M4
else
  $\tilde{\kappa}, \tilde{\chi} \leftarrow \textsc{RankNorm}(\hat{\kappa}, \hat{\chi})$
  $S(f) \leftarrow \tilde{\kappa}(f) \cdot \tilde{\chi}(f)$
end if
$(f_h, f_s) \leftarrow \text{top-2}_{f \in \mathcal{F}}\, S(f)$ ▷ tiebreak by $\hat{\chi}$ when $|S(f_h) - S(f_s)| < 0.05$
if $\hat{\chi}(f_h) = 0$ **then**
  return (t_0, v_0) ▷ abort aggregation
end if
$\tilde{t} \leftarrow \textsc{ShapeText}(t_0, f_h, f_s)$ ▷ M5: delete + reorder
$\lambda \leftarrow \hat{\chi}(f_s)/\hat{\chi}(f_h)$
for $\theta \in \mathcal{T}$ **do**
  $\text{score}(\theta) \leftarrow S(f_h; \tilde{t}, T_\theta(v_0)) + \lambda \cdot S(f_s; \tilde{t}, T_\theta(v_0))$ ▷ M6, Eq. (6)
end for
$\theta^* \leftarrow \arg\max_\theta \text{score}(\theta)$; $\tilde{v} \leftarrow T_{\theta^*}(v_0)$
if Phase 1–3 of M7 pass on $(\tilde{t}, \tilde{v})$ **then** ▷ M7 verification
  return $(\tilde{t}, \tilde{v})$
else if Phase 4 yields $(\tilde{t}_{\text{fb}}, v_0)$ passing Phases 1–3 **then**
  return $(\tilde{t}_{\text{fb}}, v_0)$
else
  return (t_0, v_0) ▷ abort to clean baseline
end if

6 Experimental Evaluation

6.1 Experimental Setup

Datasets and Victim Models. We evaluate on three multimodal recommendation benchmarks: **MicroLens-100K** [39], a Chinese short-video corpus, and **Amazon Beauty** and **Amazon Sports** [40]. Following prior work [3, 30], we use pre-extracted CLIP-ViT-B/32 visual embeddings [32] and leave-one-out evaluation. The primary victim is **VIP5** [3], a candidate-aware MLLM-RecSys built on a frozen T5-small backbone [41] and fine-tuned via PEFT adapters [42]. For cross-architecture transfer we additionally use **Qwen2-VL-2B-Instruct** [43] under few-shot and zero-shot prompts.

Baselines. We compare C3H against fifteen representative attack baselines across three categories. **Interaction-level attacks**: RandAttack [18] and PopAttack [19]. **Textual attacks**: TextFooler [20], DeepWordBug [22], BERT-Attack [21], TextBugger [23], CheckList [24], and CheatAgent [29]. **Image attacks**: EXPA, INSA, SEMA [14], VAA [44], IPDGI [25], UAP [26], and AdvCLIP [27].

Evaluation Metrics. **Effectiveness**: Exposure Rate (ER@K), the fraction of users for whom i^* appears in the top-K list. **Stealthiness**: ROUGE-$1/2/L$ [45] for textual fidelity, and the semantic-alignment gap SA Δ (absolute change in CLIP-ViT-B/32 image-text cosine similarity relative to the

Table 2: Main results of attack effectiveness and recommendation utility on **MicroLens** and **Beauty** under Few-Shot and Zero-Shot prompts. Best ER per cell is **bolded and underlined**.

Attack	*Few-Shot (MicroLens)*						*Zero-Shot (MicroLens)*					
	Top-10			**Top-20**			**Top-10**			**Top-20**		
	HR	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**
NoAttack	0.2530	0.1210	0.0000	0.4320	0.1657	0.0000	0.2510	0.1227	0.0000	0.4310	0.1679	0.0000
DeepWordBug	0.2637	0.1331	0.0360	0.4458	0.1786	0.1776	0.2627	0.1349	0.0573	0.4501	0.1817	0.1854
TextFooler	0.2568	0.1263	0.0263	0.4363	0.1712	0.1671	0.2706	0.1352	0.0359	0.4523	0.1808	0.1679
BertAttack	0.2564	0.1265	0.0343	0.4391	0.1721	0.1621	0.2626	0.1318	0.0469	0.4474	0.1781	0.1523
CheatAgent	0.2509	0.1238	0.0319	0.4412	0.1715	0.1894	0.2574	0.1290	0.0184	0.4482	0.1768	0.1631
TextBugger	0.2620	0.1316	0.0306	0.4432	0.1770	0.1973	0.2591	0.1283	0.0381	0.4494	0.1760	0.1700
CheckList	0.2599	0.1295	0.0278	0.4371	0.1737	0.1734	0.2594	0.1306	0.0366	0.4398	0.1757	0.1797
RandAttack	0.2568	0.1281	0.0296	0.4428	0.1746	0.1731	0.2532	0.1261	0.0400	0.4499	0.1753	0.1679
PopAttack	0.2607	0.1282	0.0328	0.4388	0.1727	0.1673	0.2568	0.1329	0.0424	0.4414	0.1791	0.1513
EXPA	0.2600	0.1250	0.0326	0.4456	0.1713	0.1662	0.2654	0.1334	0.0442	0.4454	0.1784	0.1628
INSA	0.2477	0.1231	0.0231	0.4347	0.1698	0.1657	0.2631	0.1328	0.0350	0.4441	0.1780	0.1588
SEMA	0.2636	0.1319	0.0328	0.4504	0.1787	0.1664	0.2670	0.1331	0.0454	0.4512	0.1793	0.1638
VAA	0.2546	0.1262	0.0318	0.4356	0.1714	0.1579	0.2561	0.1284	0.0436	0.4446	0.1755	0.1569
IPDGI	0.2636	0.1339	0.0327	0.4447	0.1792	0.1593	0.2606	0.1309	0.0462	0.4484	0.1779	0.1567
UAP	0.2518	0.1256	0.0308	0.4426	0.1734	0.1748	0.2571	0.1300	0.0450	0.4471	0.1777	0.1766
AdvCLIP	0.2578	0.1267	0.0182	0.4423	0.1729	0.1852	0.2550	0.1231	0.0321	0.4446	0.1706	0.1896
C3H	0.2554	0.1268	**0.0782**	0.4366	0.1721	**0.2727**	0.2488	0.1263	**0.0643**	0.4390	0.1741	**0.2739**
	Few-Shot (Beauty)						*Zero-Shot (Beauty)*					
	Top-10			**Top-20**			**Top-10**			**Top-20**		
NoAttack	0.3530	0.2012	0.0000	0.4870	0.2350	0.0000	0.3210	0.1829	0.0000	0.4570	0.2172	0.0000
DeepWordBug	0.3368	0.1893	0.0024	0.4808	0.2255	0.0144	0.3075	0.1681	0.0091	0.4461	0.2030	0.0306
TextFooler	0.3433	0.1935	0.0027	0.4837	0.2287	0.0166	0.3083	0.1713	0.0083	0.4456	0.2059	0.0286
BertAttack	0.3379	0.1903	0.0029	0.4768	0.2253	0.0154	0.3041	0.1673	0.0079	0.4436	0.2024	0.0288
CheatAgent	0.3464	0.1949	0.0024	0.4872	0.2304	0.0091	0.3113	0.1715	0.0064	0.4515	0.2068	0.0159
TextBugger	0.3370	0.1901	0.0017	0.4754	0.2249	0.0115	0.3080	0.1657	0.0089	0.4419	0.1993	0.0247
CheckList	0.3431	0.1947	0.0024	0.4834	0.2300	0.0116	0.3097	0.1680	0.0051	0.4453	0.2021	0.0231
RandAttack	0.3436	0.1943	0.0027	0.4885	0.2308	0.0124	0.3189	0.1737	0.0042	0.4493	0.2065	0.0167
PopAttack	0.3441	0.1954	0.0026	0.4822	0.2302	0.0149	0.3027	0.1675	**0.0110**	0.4415	0.2024	0.0351
EXPA	0.3355	0.1891	0.0020	0.4808	0.2255	0.0136	0.3053	0.1662	0.0075	0.4417	0.2005	0.0277
INSA	0.3408	0.1934	0.0029	0.4836	0.2293	0.0153	0.3034	0.1650	0.0082	0.4444	0.2005	0.0303
SEMA	0.3442	0.1947	0.0020	0.4829	0.2297	0.0140	0.3021	0.1620	0.0082	0.4329	0.1949	0.0254
VAA	0.3404	0.1940	0.0024	0.4839	0.2300	0.0152	0.3078	0.1690	0.0071	0.4496	0.2047	0.0276
IPDGI	0.3446	0.1956	0.0024	0.4864	0.2313	0.0159	0.3103	0.1694	0.0063	0.4450	0.2033	0.0254
UAP	0.3473	0.1953	0.0023	0.4929	0.2320	0.0151	0.3103	0.1695	0.0078	0.4379	0.2016	0.0290
AdvCLIP	0.3424	0.1918	0.0027	0.4806	0.2266	0.0145	0.3034	0.1654	0.0076	0.4432	0.2006	0.0313
C3H	0.3415	0.1917	**0.0049**	0.4823	0.2271	**0.0290**	0.3043	0.1657	0.0081	0.4399	0.1998	**0.0623**

clean item) for cross-modal coherence. **Utility**: HR@K and NDCG@K [46] on benign users. We use $K \in \{10, 20\}$ throughout.

Implementation Details. We implement all models in PyTorch 1.13.1 (CUDA 11.7) and fine-tune VIP5 via AdamW [47]. The VIP5 victim is trained independently for each of MicroLens, Beauty, and Sports for 20 epochs at learning rate 1×10^{-3}, effective batch size 128, warmup ratio 0.1, and gradient clipping at 5.0; each fine-tuning run takes approximately 10 hours on 4$\times$ NVIDIA L20 GPUs. The Qwen2-VL-2B victim is evaluated under few-shot and zero-shot prompts on the same hardware. The C3H pipeline is deterministic and runs in approximately 8 seconds per target when distributed across 4 GPUs ($\sim$34 s single-GPU equivalent), with no queries to the victim during attack construction. For each (target, prompt) setting, we fix a single evaluation-user subset that is shared across NoAttack and all compared attack methods: VIP5 settings score 1,000 benign users each, and Qwen2-VL settings subsample users to limit inference cost.

6.2 Overall Performance Evaluation

Attack Effectiveness. At K=20 on VIP5 (Tab. 2), C3H leads in all four (dataset, prompt) cells. On MicroLens (N=9), C3H raises ER@20 to 0.273 (FS) / 0.274 (ZS), about 40% above the strongest baseline (TextBugger 0.197 FS; AdvCLIP 0.190 ZS). Beauty is harder for content-only attacks

Table 3: Attack stealthiness on **MicroLens** (N=9) and **Beauty** (N=14). ROUGE uses character-level tokenisation on MicroLens, word-level on Beauty.

	MicroLens ($N = 9$)				*Beauty* ($N = 14$)			
	Unimodal Naturalness ROUGE			*Cross-modal*	*Unimodal Naturalness ROUGE*			*Cross-modal*
Attack	**-1↑**	**-2↑**	**-L↑**	**SA Δ ↓**	**-1↑**	**-2↑**	**-L↑**	**SA Δ ↓**
NoAttack	100.00	100.00	100.00	0.0000	100.00	100.00	100.00	0.0000
C3H	94.02	93.86	94.02	**0.0020**	82.22	80.25	82.22	**0.0135**

Table 4: Leave-one-out ablation on **MicroLens** (N=9) and **Beauty** (N=14) at K=20. ΔER is the drop relative to full C3H.

	Few-Shot (MicroLens)				*Zero-Shot (MicroLens)*				*Few-Shot (Beauty)*				*Zero-Shot (Beauty)*			
Variant	**HR**	**NDCG**	**ER**	**ΔER**	**HR**	**NDCG**	**ER**	**ΔER**	**HR**	**NDCG**	**ER**	**ΔER**	**HR**	**NDCG**	**ER**	**ΔER**
C3H (Full C3H)	0.4366	0.1721	**0.2727**	–	0.4390	0.1741	**0.2739**	–	0.4823	0.2271	**0.0290**	–	0.4399	0.1998	**0.0623**	–
−M5 (image only)	0.4430	0.1659	0.1623	−0.110	0.4488	0.1761	0.1668	−0.107	0.4834	0.2296	0.0202	−0.009	0.4500	0.2025	0.0362	−0.026
−M6 (text only)	0.4430	0.1659	0.1824	−0.090	0.4461	0.1746	0.1814	−0.092	0.4747	0.2247	0.0131	−0.016	0.4456	0.2035	0.0148	−0.048
−C2C (random hero)	0.4423	0.1738	0.1732	−0.099	0.4472	0.1772	0.1761	−0.098	0.4875	0.2306	0.0196	−0.009	0.4494	0.2051	0.0371	−0.025
NoAttack	0.4320	0.1657	0.0000	−0.273	0.4310	0.1679	0.0000	−0.274	0.4870	0.2350	0.0000	−0.029	0.4570	0.2172	0.0000	−0.062

(N=14), but C3H still reaches 0.029 (FS) / 0.062 (ZS), close to twice the runner-up. The near-identical MicroLens FS/ZS scores (0.273 vs. 0.274) suggest the C2C surrogate latches onto structural decision criteria rather than prompt-specific features. On Beauty, ZS exceeds FS (0.062 vs. 0.029): without exemplars to anchor the model's ranking, criteria-aligned content faces less competition.

Stealthiness and Utility. HR@K and NDCG@K on benign users stay close to NoAttack across all four cells of Tab. 2. Stealthiness holds on both axes (Tab. 3): the edited title keeps a high ROUGE-L with the original (94.02 on MicroLens, 82.22 on Beauty), and SA Δ stays at 0.0020 / 0.0135.

6.3 Ablation Study

We run a leave-one-out ablation (Tab. 4) at K=20, removing one component at a time: (1) the text operator (image-only); (2) the image operator (text-only); and (3) the C2C joint scoring, replaced by random hero selection from the same pool $\mathcal{F}$.

Dominant modality flips with the dataset. On text-rich MicroLens captions, removing the text operator costs more ER than removing the image operator (−0.110/−0.107 vs. −0.090/−0.092 on FS/ZS). Beauty's short titles reverse the order: dropping the image operator costs −0.016/−0.048 while the text operator costs only −0.009/−0.026. Longer text gives the text edit more to amplify; sparse titles shift the burden to the visual channel.

Joint scoring beats random hero. Replacing $\tilde{\kappa} \cdot \tilde{\chi}$ with a random pick from the same pool drops ER everywhere (−0.099/−0.098 on MicroLens; −0.009/−0.025 on Beauty), echoing §4: pool filtering pre-selects cross-modally aligned cues, and the joint score adds within-pool ranking signal on top. Utility stays within 0.02 of full C3H on every variant.

6.4 Defence Robustness

We apply three content-filter defences ahead of the recommender, each calibrated to 5–10% rejection on clean items: AE Filter [48, 49] (autoencoder latent denoising in CLIP space), MCM (Maximum Concept Matching) [50], and Mahalanobis OOD detection [51]. None meaningfully reduces C3H (Tab. 5): ER moves by less than 0.004 under every defence in all four cells, and HR@K / NDCG@K on benign users are preserved. The reason is structural: the text operator only deletes and reorders tokens already in t_0, and image transforms in $\mathcal{T}$ stay within the crops and lighting adjustments common in product photography. The edited content sits inside the clean distribution in CLIP feature space (SA $\Delta = 0.002$ on MicroLens), giving outlier-based filters nothing to latch onto.

6.5 Cross-architecture Transferability

We take the attack content $(\tilde{t}_{i^*}, \tilde{v}_{i^*})$ produced against VIP5 and apply it unchanged to Qwen2-VL-2B-Instruct [43] under few-shot and zero-shot prompts. The two victims use different vision (Qwen's

Table 5: Defence robustness on **MicroLens** (N=9) under Few-Shot and Zero-Shot prompts. Each defence is calibrated to 5–10% rejection on clean items.

Defence	*Few-Shot (MicroLens)*						*Zero-Shot (MicroLens)*					
	Top-10			**Top-20**			**Top-10**			**Top-20**		
	HR	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**
No Defence	0.2554	0.1268	0.0782	0.4366	0.1721	0.2727	0.2488	0.1263	0.0643	0.4390	0.1741	0.2739
AE Filter [48, 49]	0.2571	0.1285	0.0794	0.4426	0.1748	0.2732	0.2630	0.1319	0.0638	0.4453	0.1775	0.2718
MCM [50]	0.2604	0.1314	0.0814	0.4510	0.1793	0.2736	0.2650	0.1312	0.0628	0.4471	0.1768	0.2747
Mahalanobis OOD [51]	0.2603	0.1301	0.0801	0.4479	0.1771	0.2714	0.2544	0.1278	0.0634	0.4420	0.1748	0.2709

native ViT vs. CLIP-ViT-B/32) and language (1.5B Qwen-2 LM vs. T5-small) backbones. Full tables are in App. B; VIP5 Sports results are in App. A.

C3H ranks first in 4 of 6 (dataset, prompt) cells at K=20 on Qwen2-VL, including all three Zero-Shot cells. Sports is the clearest case: C3H wins both prompt settings by 3.7–4.7× over the runner-up. The Zero-Shot > Few-Shot pattern from VIP5 also reappears: at K=20, C3H reaches 0.117 (ZS) vs. 0.054 (FS) on MicroLens, 0.101 vs. 0.056 on Beauty, and 0.187 vs. 0.113 on Sports. The mechanism does not hinge on the specific victim. On Few-Shot MicroLens and Beauty, a few image-attack baselines edge ahead of C3H; we attribute this to large pixel perturbations disturbing Qwen2-VL's in-context reasoning when few-shot exemplars are present, which is unrelated to the C2C bias and does not generalise to Sports (the same baselines drop below ER 0.03 on every Few-Shot cell there).

7 Conclusion

We formalise and study decision criteria hijacking against MLLM-RecSys under a strict black-box, zero-query, inference-time, single-seller threat model. Candidate-aware MLLM-RecSys exhibit a compression-to-consensus (C2C) bias: the same compression and cross-modal fusion mechanisms that suppress conventional unimodal perturbations also confer disproportionate ranking influence on cues that survive compression and receive cross-modal reinforcement. C3H exploits this bias as a query-free, public-encoder-driven pipeline that edits a single target item's text and image inside a stealthiness set $\mathcal{B}$.

On VIP5, C3H beats fifteen baselines on every (dataset, prompt) cell at ER@20 by 1.1–1.8×, and the same content transfers unchanged to Qwen2-VL-2B-Instruct, winning all Zero-Shot cells across MicroLens, Beauty, and Sports, with margins reaching 4.7× over the runner-up on Sports. Three anomaly-based content-filter defences fail to mitigate C3H ($\Delta\mathrm{ER} < 0.004$), and benign-user utility (HR@K, NDCG@K) is preserved. Compression and cross-modal fusion are commonly assumed to confer robustness on MLLM-RecSys; our results show that they also open an on-criterion attack surface that requires no victim feedback.

8 Limitations and Potential Mitigations

C3H assumes edit access to both text and images, and its effectiveness may vary across targets, victim models, and platform constraints. Our evaluation covers a limited set of targets and victim models in controlled benchmarks; effectiveness on deployed platforms remains to be established. Candidate mitigations include content canonicalisation and monitoring listing edits, but their effectiveness against C3H requires further evaluation, and they may reject legitimate edits or reduce recommendation utility.

Acknowledgements

This research / project is supported by A*STAR under its Japan-Singapore Joint Call: JST-A*STAR 2024 (Project ID: R2416IR139). This research is also supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award MOE-T2EP20125-0005). The authors declare no competing interests.

References

[1] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[2] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. VIP5: Towards multimodal foundation models for recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9606–9620, 2023.

[4] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[5] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4):1–19, 2015.

[6] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198, 2016.

[7] Yurong Hao, Wen Zhou, Guowei Guan, Tiantong Wu, Fuyao Zhang, and Wei Yang Bryan Lim. VirusCascade: Hijacking collaborative reflection in LLM-powered recommender agents. *arXiv preprint arXiv:2609.38270*, 2026.

[8] Ramin Giahi, Kehui Yao, Sriram Kollipara, Kai Zhao, Vahid Mirjalili, Jianpeng Xu, Topojoy Biswas, Evren Korpeoglu, and Kannan Achan. VL-CLIP: Enhancing multimodal recommendations via visual grounding and LLM-augmented CLIP embeddings. In *Proceedings of the 19th ACM Conference on Recommender Systems*, pages 482–491, 2025.

[9] Yuyang Ye, Zhi Zheng, Yishan Shen, Tianshu Wang, Hengruo Zhang, Peijun Zhu, Runlong Yu, Kai Zhang, and Hui Xiong. Harnessing multimodal large language models for multimodal sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13069–13077, 2025.

[10] Fuyao Zhang, Jiaming Zhang, Che Wang, Boyang Chen, Yurong Hao, Xiongtao Sun, Guowei Guan, Blaise Delattre, Yang Cao, and Wei Yang Bryan Lim. SynChain: Inducing computer-use agent systems to construct their own attack chains. *arXiv preprint arXiv:2608.06862*, 2026.

[11] Wenkuan Zhao, Shanshan Zhong, Yifan Liu, Wushao Wen, Jinghui Qin, Mingfu Liang, and Zhongzhan Huang. DVIB: Towards robust multimodal recommender systems via variational information bottleneck distillation. In *Proceedings of the ACM on Web Conference 2025*, pages 2549–2561, 2025.

[12] Guowei Guan, Yurong Hao, Jiaming Zhang, Tiantong Wu, Fuyao Zhang, Tianxiang Chen, Longtao Huang, Cyril Leung, and Wei Yang Bryan Lim. VENOMREC: Cross-modal interactive poisoning for targeted promotion in multimodal LLM recommender systems. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 37417–37433. PMLR, 2026.

[13] Yuancheng Xu, Jiarui Yao, Manli Shu, Yanchao Sun, Zichu Wu, Ning Yu, Tom Goldstein, and Furong Huang. Shadowcast: Stealthy data poisoning attacks against vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 57733–57764, 2024.

[14] Zhuoran Liu and Martha Larson. Adversarial item promotion: Vulnerabilities at the core of top-N recommenders that use images to address cold start. In *Proceedings of the Web Conference 2021*, pages 3590–3602, 2021.

[15] Yurong Hao, Xihui Chen, Xiaoting Lyu, Jiqiang Liu, Yongsheng Zhu, Zhiguo Wan, Sjouke Mauw, and Wei Wang. Not one less: Exploring interplay between user profiles and items in untargeted attacks against federated recommendation. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2889–2903, 2024.

[16] Yurong Hao, Xihui Chen, Wei Wang, Jiqiang Liu, Tao Li, Junyong Wang, and Witold Pedrycz. Eyes on federated recommendation: Targeted poisoning with competition and its mitigation. *IEEE Transactions on Information Forensics and Security*, 19:10173–10188, 2024.

[17] Yaoyao Qian, Yifan Zeng, Yuchao Jiang, Chelsi Jain, and Huazheng Wang. The ranking blind spot: Decision hijacking in LLM-based text ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21958–21968, 2025.

[18] Michael P. O'Mahony, Neil J. Hurley, and Guénolé C. M. Silvestre. Recommender systems: Attack types and strategies. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 334–339, 2005.

[19] Wei Zhou, Junhao Wen, Min Gao, Ling Liu, Haini Cai, and Xibin Wang. A shilling attack detection method based on SVM and target item analysis in collaborative filtering recommender systems. In *Proceedings of the 8th International Conference on Knowledge Science, Engineering and Management*, pages 751–763. Springer, 2015.

[20] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8018–8025, 2020.

[21] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6193–6202, 2020.

[22] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops*, pages 50–56. IEEE, 2018.

[23] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. TextBugger: Generating adversarial text against real-world applications. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. The Internet Society, 2019.

[24] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.

[25] Lijian Chen, Wei Yuan, Tong Chen, Guanhua Ye, Nguyen Quoc Viet Hung, and Hongzhi Yin. Adversarial item promotion on visually-aware recommender systems by guided diffusion. *ACM Transactions on Information Systems*, 42(6):1–26, 2024.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[27] Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. AdvCLIP: Downstream-agnostic adversarial examples in multimodal contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6311–6320, 2023.

[28] Fuyao Zhang, Jiaming Zhang, Che Wang, Xiongtao Sun, Yurong Hao, Guowei Guan, Wenjie Li, Longtao Huang, and Wei Yang Bryan Lim. DualTAP: A dual-task adversarial protector for mobile MLLM agents. In *Proceedings of the 19th European Conference on Computer Vision*, pages 323–340. Springer, 2026.

[29] Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. CheatAgent: Attacking LLM-empowered recommender systems via LLM agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2284–2295, 2024.

[30] Jinghao Zhang, Yuting Liu, Qiang Liu, Shu Wu, Guibing Guo, and Liang Wang. Stealthy attack on large language model based recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5839–5857, 2024.

[31] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.

[32] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[33] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

[34] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

[35] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377. University of Illinois, 1999.

[36] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, 2020.

[37] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2793–2803. PMLR, 2021.

[38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[39] Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. A content-driven micro-video recommendation dataset at scale. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6486–6491, 2025.

[40] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172, 2013.

[41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[42] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-Adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022.

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[44] Rami Cohen, Oren Sar Shalom, Dietmar Jannach, and Amihood Amir. A black-box attack model for visually-aware recommender systems. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 94–102, 2021.

[45] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

[46] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[48] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[49] Celia Cintas, Skyler Speakman, Victor Akinwande, William Ogallo, Komminist Weldemariam, Srihari Sridharan, and Edward McFowland. Detecting adversarial attacks via subset scanning of autoencoder activations and reconstruction error. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 876–882. International Joint Conferences on Artificial Intelligence Organization, 2020.

[50] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. In *Advances in Neural Information Processing Systems*, volume 35, pages 35087–35102, 2022.

[51] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7167–7177, 2018.

Appendix

A Results on Sports Dataset

We additionally report VIP5 results on the Sports dataset (N=6 attackable targets) for completeness. Tab. 6 shows that C3H remains the strongest attack at K=20 on both prompts (FS: 0.028 vs. best baseline 0.021; ZS: 0.106 vs. 0.096), though the absolute margin is smaller than on MicroLens and Beauty due to the smaller attackable subset.

Table 6: Main results on **Sports** (N=6) under Few-Shot and Zero-Shot prompts at K=20.

	Few-Shot (Sports)			*Zero-Shot (Sports)*		
Attack	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**
NoAttack	0.3240	0.1666	0.0000	0.3210	0.1526	0.0000
DeepWordBug	0.3397	0.1590	0.0095	0.3373	0.1542	0.0755
TextFooler	0.3470	0.1627	0.0117	0.3305	0.1515	0.0937
BertAttack	0.3418	0.1605	0.0212	0.3290	0.1490	0.0808
CheatAgent	0.3385	0.1600	0.0073	0.3420	0.1534	0.0338
TextBugger	0.3385	0.1572	0.0070	0.3315	0.1541	0.0728
CheckList	0.3482	0.1674	0.0153	0.3338	0.1569	0.0692
RandAttack	0.3435	0.1594	0.0047	0.3293	0.1547	0.0728
PopAttack	0.3387	0.1569	0.0083	0.3282	0.1486	0.0745
EXPA	0.3522	0.1615	0.0177	0.3428	0.1560	0.0928
INSA	0.3420	0.1576	0.0132	0.3342	0.1527	0.0838
SEMA	0.3463	0.1606	0.0182	0.3348	0.1541	0.0962
VAA	0.3328	0.1539	0.0065	0.3463	0.1579	0.0618
IPDGI	0.3447	0.1604	0.0133	0.3425	0.1559	0.0745
UAP	0.3398	0.1605	0.0048	0.3400	0.1568	0.0508
AdvCLIP	0.3422	0.1572	0.0100	0.3270	0.1487	0.0702
C3H	0.3395	0.1598	**0.0282**	0.3380	0.1549	**0.1058**

B Cross-Architecture Transferability: Results

The cross-architecture transferability of C3H on Qwen2-VL-2B-Instruct, summarised in §6.5, is reported in Tab. 7 (MicroLens, N=9; Beauty, N=14) and Tab. 8 (Sports, N=6) under both Few-Shot and Zero-Shot prompts.

Table 7: Cross-architecture transferability on **Qwen2-VL-2B-Instruct** [43] for **MicroLens** (N=9) and **Beauty** (N=14) at K=20.

Attack	**MicroLens** (N=9)						**Beauty** (N=14)					
	Few-Shot			*Zero-Shot*			*Few-Shot*			*Zero-Shot*		
	HR	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**
NoAttack	0.3144	0.1033	0.0000	0.2534	0.0809	0.0000	0.3583	0.1563	0.0000	0.3908	0.1685	0.0000
DeepWordBug	0.3229	0.1073	0.0781	0.2398	0.0828	0.0272	0.3610	0.1491	0.0475	0.4032	0.1776	0.0194
TextFooler	0.3081	0.0978	0.0514	0.2423	0.0749	0.0223	0.3410	0.1417	0.0033	0.4149	0.1772	0.0035
BertAttack	0.3306	0.1053	0.0376	0.2742	0.0950	0.0055	0.3295	0.1361	0.0300	0.4124	0.1791	0.0053
CheatAgent	0.3173	0.1043	0.0533	0.2472	0.0822	0.0306	0.3558	0.1584	0.0066	0.4021	0.1830	0.0053
TextBugger	0.3193	0.1075	0.0765	0.2287	0.0819	0.0055	0.3378	0.1422	0.0532	0.3936	0.1619	0.0160
CheckList	0.3333	0.1060	0.0355	0.2316	0.0779	0.0327	0.3523	0.1518	0.0417	0.3929	0.1649	0.0336
RandAttack	0.3306	0.1036	0.0190	0.2310	0.0789	0.0027	0.3250	0.1461	0.0536	0.3838	0.1683	0.0018
PopAttack	0.3253	0.1057	0.0269	0.2830	0.0874	0.0220	0.3673	0.1582	0.0000	0.4142	0.1762	0.0017
EXPA	0.2992	0.0970	0.0943	0.2646	0.0905	0.0056	0.3289	0.1488	0.0548	0.4139	0.1685	0.0226
INSA	0.3152	0.1035	0.0870	0.2472	0.0799	0.0083	0.3356	0.1430	0.0688	0.4152	0.1732	0.0247
SEMA	0.2861	0.0960	**0.0989**	0.2231	0.0735	0.0028	0.3384	0.1429	0.0643	0.4004	0.1735	0.0212
VAA	0.3431	0.1122	0.0878	0.2630	0.0831	0.0055	0.3267	0.1424	0.0783	0.3878	0.1724	0.0174
IPDGI	0.3173	0.1000	0.0907	0.2707	0.0891	0.0083	0.3261	0.1318	0.0765	0.3748	0.1597	0.0294
UAP	0.3154	0.1021	0.0755	0.2472	0.0847	0.0000	0.3570	0.1480	**0.0959**	0.3537	0.1628	0.0470
AdvCLIP	0.3592	0.1156	0.0858	0.2598	0.0865	0.0000	0.3710	0.1666	0.0682	0.4207	0.1788	0.0414
C3H	0.3029	0.0959	0.0536	0.2542	0.0833	**0.1173**	0.3237	0.1420	0.0559	0.4108	0.1738	**0.1014**

Table 8: Cross-architecture transferability on **Qwen2-VL-2B-Instruct** [43] for **Sports** (N=6) at K=20.

	Few-Shot (Sports)			*Zero-Shot (Sports)*		
Attack	**HR**	**NDCG**	**ER**	**HR**	**NDCG**	**ER**
NoAttack	0.3434	0.1517	0.0000	0.2877	0.1307	0.0000
DeepWordBug	0.3352	0.1481	0.0166	0.3271	0.1489	0.0141
TextFooler	0.3481	0.1576	0.0055	0.2941	0.1358	0.0118
BertAttack	0.3111	0.1536	0.0000	0.3129	0.1393	0.0000
CheatAgent	0.3225	0.1443	0.0000	0.3002	0.1412	0.0000
TextBugger	0.3011	0.1272	0.0304	0.3170	0.1371	0.0326
CheckList	0.2869	0.1266	0.0301	0.3333	0.1481	0.0399
RandAttack	0.2779	0.1180	0.0027	0.3155	0.1432	0.0093
PopAttack	0.3171	0.1325	0.0054	0.3121	0.1430	0.0000
EXPA	0.2877	0.1322	0.0247	0.3192	0.1380	0.0000
INSA	0.2992	0.1234	0.0216	0.3357	0.1496	0.0188
SEMA	0.3306	0.1506	0.0246	0.3019	0.1352	0.0024
VAA	0.3562	0.1543	0.0274	0.3341	0.1519	0.0000
IPDGI	0.2981	0.1225	0.0163	0.3341	0.1484	0.0141
UAP	0.3115	0.1245	0.0273	0.3095	0.1437	0.0000
AdvCLIP	0.3062	0.1291	0.0108	0.3066	0.1356	0.0000
C3H	0.3113	0.1430	**0.1129**	0.2937	0.1261	**0.1865**

Hyperparameters used in our experiments. For all reported results, we use $\tau_{\text{cat}} = 0.20$, $\tau_{\text{orig}} = 0.18$, $\tau_{\text{align}} = 0.18$ (B/32 cosine thresholds), context window lengths $L_{\text{text}} \in \{20, 40, 80\}$ for TextRank, top-$\lceil P/4 \rceil$ patches for $V(f)$ under attention rollout, position weights $\{1.0, 0.5, 0.2\}$ for title/body/implicit in A_{text}, and tie-break threshold $\Delta_{\text{tie}} = 0.05$ for hero/secondary selection. Image transform parameters: θ_1, θ_2 use Grad-CAM on CLIP-ViT-B/32, with the crop bounding box expanded by 20% around the cue saliency centroid; θ_3 centre-crops to $0.7\times$ each side; θ_4 applies contrast factor 1.15 and brightness shift $+0.05$ within the hero region (Grad-CAM mask thresholded at the 50th percentile). Small-image fallback: $\mathcal{T} = \{\theta_0, \theta_4\}$ when image side < 200 px, $\mathcal{T} = \{\theta_0\}$ when < 100 px.

C Compute Reporting

We report the computational resources used for constructing and evaluating C3H. The paper reports results on three datasets (MICROLENS, Amazon BEAUTY, and Amazon SPORTS); the compute estimates below also include additional exploratory experiments beyond these reported results. We fine-tune the VIP5 victim with parameter-efficient adapters, generate C3H content for each (dataset, target) pair using only public encoders, and evaluate effectiveness, utility, and stealthiness on benign user pools. No models are trained from scratch and no external API services are used; all inference is performed locally.

Local Compute Environment. All experiments run on a single Ubuntu 20.04 workstation with $8\times$ NVIDIA L20 48 GB GPUs (CUDA 11.7), Python 3.9, and PyTorch 1.13.1+cu117. Source code, checkpoints, evaluation logs, and pre-extracted CLIP features occupy approximately 90 GB of disk storage.

Victim Fine-Tuning. We fine-tune the VIP5 victim via VL-Adapter [42] (reduction factor 8) for 20 epochs per dataset using AdamW at learning rate 1×10^{-3}, warmup ratio 0.1, gradient clipping 5.0, an effective batch size of 128 (4×32 across four L20 GPUs), and a fixed random seed of 2022. Each dataset's fine-tuning takes approximately 9.5–10.5 wall-clock hours; across the three datasets reported in the paper (MICROLENS, Amazon BEAUTY, and Amazon SPORTS) and additional exploratory fine-tuning, victim fine-tuning consumes approximately **200** GPU-hours.

Attack Generation. Each C3H artefact is generated once per (dataset, target) pair using only public CLIP-ViT-B/16 (patch-level for $\hat{\kappa}$) and CLIP-ViT-B/32 (global for $\hat{\chi}$) encoders. Generation is deterministic, requires no victim queries, and takes approximately 30–40 seconds per target on a single L20 GPU. Aggregated across all content-modification baselines and C3H ablation variants, attack generation consumes approximately **30** GPU-hours.

Evaluation Compute. Effectiveness and utility metrics are computed via VIP5's pointwise evaluator (B-3 few-shot and B-Z zero-shot prompts, 99-candidate pool) on $1{,}000$ benign users per (target, attack, prompt) cell. Each cell costs roughly 3–4 minutes of L20 GPU time. The reported main results, ablations, defence evaluations, and mechanism analysis, together with additional exploratory evaluations, comprise $6{,}213$ pointwise evaluation runs, totalling approximately $\mathbf{345}$ GPU-hours. For cross-architecture transferability, we evaluate Qwen2-VL-2B-Instruct [43] with frozen parameters (`bfloat16` inference, no fine-tuning, no LoRA), under both *few-shot* (two in-context demonstrations) and *zero-shot* prompt regimes that mirror the VIP5 protocol. The six evaluation jobs (3 datasets $\times$ 2 prompt regimes) run in parallel across six L20 GPUs, completing in 13–32 wall-clock minutes each and contributing approximately $\mathbf{3}$ GPU-hours.

Total Compute, Retries, and Notes. Summing across phases, the full C3H study consumes approximately $\mathbf{580}$ NVIDIA L20 GPU-hours; this figure includes additional exploratory experiments beyond the three datasets reported in the paper. Evaluation scripts include automatic retries on transient CUDA OOM events (observed retry rate below 1%). A small fraction of attack-generation jobs abort at the M7 stealthiness gate and return the clean baseline (t_0, v_0) as specified in Sec. 5.3; these are by-design fallbacks, not failures. No model is trained from scratch.

D Broader Impacts and Responsible Use

This work aims to improve the security of multimodal recommender systems by identifying vulnerabilities in content-based ranking. The findings may help researchers and platform operators develop more robust recommendation mechanisms. However, C3H is dual-use: similar techniques could be misused by malicious sellers to manipulate product exposure, potentially misleading users and disadvantaging other sellers. Our experiments use public benchmark datasets and research models; we do not attack deployed commercial platforms. Monitoring listing edits and content canonicalisation are potential mitigations, although their effectiveness against C3H remains to be evaluated. We encourage the use of research artefacts for controlled security evaluation and do not redistribute modified benchmark media.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and §1 state the C2C bias hypothesis, the strict zero-query single-seller threat model, and the headline empirical claims; the experiments in §6 match the scope and magnitude of these claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sec. 8 discusses the attack's assumptions, the scope of the evaluation, and candidate mitigations and their trade-offs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results; the information-bottleneck arguments cited in §5 serve as motivation rather than proven contributions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: §6.1 specifies datasets, victims, baselines, and metrics; the full pipeline pseudocode is in Alg. 1; hyperparameters and small-image fallbacks are detailed in App. B; the compute environment is reported in App. C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code is publicly available at `https://github.com/GuoweiGuan666/C3H`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and evaluation details (data splits, optimizer, learning rate, batch size, seeds, prompt regimes, and user-subset sampling) are reported in §6.1 and App. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The final manuscript reports descriptive ER@K, HR@K, and NDCG@K results in the main and ablation tables, but does not report error bars, confidence intervals, or formal significance tests. Evaluation-user sampling is described in §6.1. The reported differences are interpreted as descriptive comparisons in the evaluated settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: App. C reports the hardware (NVIDIA L20 48 GB), per-phase wall-clock breakdown, and the total compute (approximately 580 GPU-hours).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and the work conforms in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix D discusses the potential security benefits, dual-use risks, and broader societal impacts of C3H, together with the scope of our evaluation and possible mitigations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release pre-trained attack models or modified benchmark datasets. The potential dual-use risks of C3H and its research artefacts are discussed in Appendix D.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and base models are cited in Sec. 6.1. The VIP5 and CLIP repositories use the MIT licence; T5 and Qwen2-VL-2B-Instruct use Apache-2.0. The MicroLens README and Amazon review-data page request citation without specifying named dataset licences. The MicroLens README restricts redistribution of modified data; its download portal additionally specifies academic-use and modification/distribution restrictions for videos. We do not redistribute original or edited MicroLens text, images, or videos; our artefact release is limited to code, configurations, and numerical results.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces C3H and provides a public repository containing source code and documentation. The main paper and appendices document the attack pipeline, experimental settings, hyperparameters, and computational resources, while the repository provides instructions for reproducing the workflow.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects; benign-user data is drawn from existing public recommendation benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The MLLM-RecSys studied are the object of evaluation, not a component of the authoring or methodology pipeline; no LLM was used in a non-standard way that affects the core methods, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.