What Do Neighbors Know? Open-World Semantic Inference Attack on Intermediate Representations

Bangjie Sun
National University of Singapore
bangjie@comp.nus.edu.sg

Sean Rui Xiang Tan
National University of Singapore
seantanr@comp.nus.edu.sg

Rui Xiao
Shanghai University of Finance and Economics
xiaorui@sufe.edu.cn

Mun Choon Chan
National University of Singapore
chanmc@comp.nus.edu.sg

Jun Han
KAIST
junhan@cyphy.kaist.ac.kr

Abstract

Deep learning systems increasingly expose intermediate representations across trust boundaries. While these representations leak unintended semantic information, existing inference attacks are fundamentally *closed-world*. An attacker must predefine the sensitive attributes to test. They cannot discover unanticipated leakage channels. This limitation obscures a critical vulnerability: the risk of open-world semantic inference. To explore this vulnerability, we present *REVEAL*, an *open-world* semantic inference attack. *REVEAL* discovers what an embedding leaks without prior assumptions. It uses *k*NN retrieval over an auxiliary corpus and LLM reasoning to surface unknown candidate attributes from the embedding's local geometry. It then uses the predicted class to quantify *excess leakage*. This metric captures the semantic information encoded strictly beyond what the model's own predictions reveal. Evaluations on a controlled synthetic dataset of 160 images across six deep learning models demonstrate that open-world excess leakage is a consistent vulnerability. *REVEAL* successfully discovers 109 unique semantic attributes exhibiting substantial excess leakage. Up to 97.6% of these discovered attributes are verified as genuinely present in the source images.

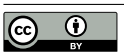
CCS Concepts

• **Security and privacy → Distributed systems security**; • **Computing methodologies → Machine learning approaches**.

Keywords

Semantic Leakage, Distributed Learning and Inference

ACM Reference Format:
Bangjie Sun, Sean Rui Xiang Tan, Rui Xiao, Mun Choon Chan, and Jun Han. 2026. *What Do Neighbors Know?* Open-World Semantic Inference Attack on Intermediate Representations. In *The 24th Annual International Conference on Mobile Systems, Applications and Services (MobiSys Workshop '26), June 21–25, 2026, Cambridge, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3812836.3815162



MobiSys Workshop '26, Cambridge, United Kingdom

ACM ISBN 979-8-4007-2712-2/26/06
https://doi.org/10.1145/3812836.3815162

1 Introduction

Modern intelligent systems increasingly rely on edge devices, such as smart cameras and IoT sensors, to process data in real time. However, running large-scale neural networks on these resource-constrained devices is impractical. Consequently, distributed intelligence paradigms like split computing and inference offloading have become essential [6, 9]. Instead of sending raw, sensitive data to the cloud, the edge device runs the initial layers of a model and transmits only the resulting ***intermediate representation*** (e.g., embeddings). While this architectural choice appears to protect user privacy by abstracting away the original data, these transmitted high-dimensional representations routinely cross trust boundaries and carry profound, unrecognized security and privacy risks [4, 12].

Prior research suggests that intermediate representations encode rich and unintended information [5, 9]. This includes demographic attributes, scene context, and other sensitive properties that an adversary could exploit [3, 4, 8]. For example, consider an image classifier deployed on a smart home camera for home security. The intended task is solely to predict whether an intruder is present. However, the transmitted image embedding could inadvertently leak the homeowner's age, gender, or the layout of the living room. This vulnerability exposes a critical attack surface in distributed AI systems: ***what semantic information can an adversary infer from an exposed representation beyond the intended task (i.e., the predicted class)?*** Existing inference attacks attempt to exploit this leakage but remain fundamentally closed-world [2–4, 7, 8]. They require the attacker to specify ***a predefined set of target attributes*** in advance. While these methods can confirm the leakage of a known trait, the semantic content of an embedding is naturally ***unknown*** a priori. An attacker cannot realistically enumerate every possible sensitive attribute beforehand. Because existing inference attacks rely entirely on closed-world testing, they cannot systematically discover unanticipated semantic channels hidden within the high-dimensional data. This shared limitation obscures an ***underexplored vulnerability: the risk of open-world semantic inference***.

To explore and evaluate this vulnerability, we present *REVEAL* (**R**epresentation **E**xploration **V**ia **E**mbedding **A**nalysis and **L**LM reasoning). *REVEAL* is an *open-world* semantic inference attack targeting intermediate representations in distributed AI systems. It operates on exposed embeddings and the model's public API and requires no access to the raw input data. Rather than relying on a predefined set of attributes, *REVEAL* treats the semantic content

of a representation as an unknown variable that must be actively discovered. We implement this through a *discover-then-measure* pipeline. To interpret a target embedding, *REVEAL* first retrieves its k nearest neighbors (kNN) from an auxiliary dataset of captioned images. Because nearby embeddings encode similar features, their text captions provide clues about the target's hidden content. *REVEAL* then uses a large language model (LLM) to analyze these retrieved captions. The LLM identifies common themes and extracts candidate semantic attributes directly from this surrounding context. Finally, *REVEAL* uses the task output (i.e., predicted class) to measure excess leakage. This step uses an information-theoretic KL decomposition to quantify the semantic information strictly beyond what the model's own predictions reveal.

To evaluate *REVEAL*, we construct a controlled synthetic dataset of 160 images using Stable Diffusion 3. Each image follows a strict composition consisting of a primary object, a secondary object, a ternary object, and a background scene. The primary object serves as the intended task label and is the most salient element. The remaining components provide unintended semantic context. We use the Densely Captioned Images (DCI) dataset [11] as the auxiliary corpus. We evaluate the attack across multiple deep learning architectures, including MobileNet variants, ResNet-50, EfficientNet, ConvNeXt-Tiny, and ViT-B. Across these models, *REVEAL* successfully surfaces 109 **unique** semantically meaningful attributes. These attributes are clearly encoded in the representation but remain completely absent from the task output. For ConvNeXt-Tiny, *REVEAL* successfully infers secondary, ternary objects, or background scenes from 39% of the target representations. We further verify the semantic validity of discovered attributes to ensure *REVEAL* extracts actual semantic leakage rather than hallucinating spurious concepts. Our evaluation shows that up to 97.6% of these attributes are genuinely present in the source images.

While this paper focuses on image classification to demonstrate the effectiveness of *REVEAL*, the attack is task-agnostic by design. It can be naturally extended to evaluate the security of other intermediate representations in emerging edge computing environments. Ultimately, our work challenges the conventional reliance on predefined threat checklists. By proposing an open-world semantic inference attack, we take the first step toward exposing the underexplored attack surface of distributed intelligence systems.

2 Background and Related Work

We review prior work on semantic inference attacks to highlight a shared closed-world assumption and an unexplored vulnerability: the risk of open-world semantic inference.

2.1 Inference Attacks on Representations

Attribute Inference. Intermediate representations encode substantially more than the intended task signal [5, 9]. This leakage naturally enables semantic attribute inference attacks [8]. However, whether exploiting latent representations [3] or using probing classifiers [1, 2], existing attacks share a fundamental constraint. They require the adversary to specify the target attribute beforehand.

Data Reconstruction. In split inference settings, adversaries often employ model inversion attacks to reconstruct the raw input image from intercepted representations [6, 10]. The goal of these attacks is strictly visual recovery. They aim for pixel-level fidelity rather than semantic attribution. They do not attempt to discover or identify specific semantic attributes leaked beyond the task signal.

Semantic Extraction from Embeddings. The closest related attack is SLImE [4], which trains a retriever to translate image embeddings back into standard text captions. While the threat model is similar, our objectives are fundamentally different. SLImE simply reconstructs expected image content using a known vocabulary. It does not actively hunt for hidden or unanticipated attributes. Furthermore, SLImE requires complex training to align different representation spaces. It also lacks a formal way to separate the information needed for the main task from genuinely excess leakage.

Research Work	Access Required	Open-World?
ML-Doctor [8]	Black-box API	No
Bukhari et al. [3]	White-box (Weights)	No
Alain et al. [1]	Representation	No
He et al. [6]	Representation	No
SLImE [4]	Representation + API	No
***REVEAL* (Ours)**	Representation + API	**Yes**

Table 1: Comparison of inference attacks. *REVEAL* is the only attack capable of open-world semantic discovery using a practical threat model.

2.2 Research Gap

Table 1 highlights the central research gap. Prior approaches consistently operate under a closed-world assumption. For instance, traditional inference attacks [1, 3, 8] only evaluate predetermined semantic attributes. Even recent semantic attacks like SLImE [4] remain constrained to a predefined vocabulary. Crucially, ***no existing attack can discover unanticipated semantic leakage*** under a practical black-box threat model. *REVEAL* addresses this gap by proposing an open-world semantic inference attack. Our discover-then-measure pipeline actively discovers unknown leaked attributes using only the exposed representation and the public API.

3 Threat Model

We formalize an open-world semantic inference attack under a practical black-box threat model. **(1) Attack Scenarios.** The adversary acts as an eavesdropper in a distributed environment. This adversary could be an *honest-but-curious edge server* covertly profiling users, a *network eavesdropper* intercepting unencrypted embeddings, or *on-device malware* sniffing the representation from client memory before transmission. **(2) Attacker Capabilities.** Regardless of the deployment, the attacker intercepts the intermediate representation transmitted across the trust boundary and queries the model's public API to prepare auxiliary corpus and obtain the final predicted output. They completely lack access to the raw input data, the model architecture, or the internal weights. **(3) Attacker Goals.** Equipped with only an auxiliary dataset of captioned images and off-the-shelf language models, the attacker's goal is twofold.

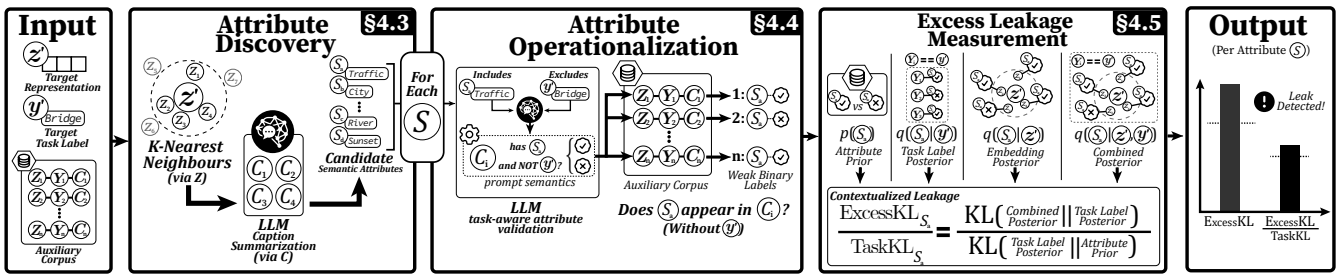


Figure 1: Overview of *REVEAL*. Our attack consists of three main stages: (i) *Discovery*: surface candidate attributes from neighborhoods in the representation space; (ii) *Operationalization*: convert the discovered attributes into weak labels that can be systematically queried; and (iii) *Excess Leakage Measurement*: quantify how much of each attribute is explained by the task versus how much remains as exploitable excess leakage.

First, they actively explore the intercepted representation to discover unknown, unintended semantic attributes encoded within the high-dimensional embedding. Second, they quantify the *excess leakage* to infer sensitive attributes that exist strictly beyond what the model's intended task prediction already reveals.

4 Design of *REVEAL*

We present the formulation and methodology of *REVEAL*.

4.1 Overview

Our goal is to discover and infer *semantic attributes* from an exposed intermediate representation. Informally, a successful attack extracts semantic content that goes ***beyond*** what is needed for the intended task. Unlike prior inference attacks (see §2), we do not assume the adversary predetermines semantic attributes in advance. Instead, our attack pipeline first ***discovers*** candidate attributes and then ***measures*** how much additional information the representation provides about them beyond the task label. The attack unfolds in three stages as depicted in Figure 1: (i) **Discovery:** surface candidate attributes from neighborhoods in the representation space; (ii) **Operationalization:** convert the discovered attributes into *weak labels*[1] that can be systematically queried; and (iii) **Excess Leakage Measurement:** quantify how much of each attribute is explained by the task versus how much remains as exploitable excess leakage.

4.2 Problem Formulation

Let z' denote the intercepted intermediate representation, and let y' denote the intended task output (i.e., predicted class) queried from the model's API. Let S denote a sensitive semantic attribute encoded in z', such as a scene, object, or contextual cue. As the adversary does not assume the attribute S beforehand, the key challenge for the attacker is therefore twofold. First, what unknown semantic attributes S are encoded in z'? Second, once S is discovered, how much exploitable information about it is present in z' strictly beyond what is already revealed by the public output y'?

[1]Weak labels are annotations generated automatically or heuristically rather than being manually assigned by a human expert.

4.3 Attribute Discovery

In the first stage, *REVEAL* actively explores the representation space for encoded semantic content instead of testing it against a predefined checklist of attributes. For each intercepted target representation z', *REVEAL* retrieves its top-k nearest neighbors from an auxiliary corpus of natural-language captions, using cosine similarity in the representation space. The core intuition is that nearby points share similar semantic structure, and their associated captions provide a semantic context for extracting shared concepts.

Then, the retrieved captions are summarized by a large language model (LLM), which acts as the reasoning engine to propose semantic attributes appearing consistently across the neighborhood. Because these attributes are generated free-form, they are not restricted to a predefined ontology. This mechanism allows *REVEAL* to surface unanticipated concepts that closed-world testing would miss.

Since not every neighborhood yields reliable signals, *REVEAL* applies simple validity checks before proceeding. Specifically, we test whether the retrieved captions are semantically coherent by computing their average pairwise *text similarity*. We also test whether the discovered attributes remain stable by measuring their *Jaccard similarity* across repeated LLM prompts with small perturbations. Only candidates passing these checks are retained for subsequent inference.

4.4 Attribute Operationalization

The first stage discovers candidate attributes as raw, natural-language text. However, a free-form phrase like "indoor setting" is too ambiguous for an adversary to systematically track across thousands of images. The second stage solves this by converting the discovered text into a searchable rule (i.e., a weak label). For each discovered attribute S, the LLM receives the neighborhood summary and the intended task label y'. We instruct the LLM to separate the core task from extra, unintended details. It then outputs a structured checklist for S containing specific keywords to find (positive cues) and keywords to ignore (exclusion cues). For example, if the primary task is identifying a "dog," the embedding might unintentionally leak that the photo was taken in an "indoor living room." To track this leaked setting, the LLM creates a rule that searches captions for terms like "couch" or "carpet" while excluding outdoor words like

"grass" or "yard." We apply these exact matching rules across the auxiliary corpus to automatically assign binary labels. This allows the attack to scale massively without human intervention.

4.5 Excess Leakage Measurement

Once an attribute S is discovered and weakly labeled, *REVEAL* then determines *how predictable* it is from the task label alone, and *how much additional predictive signal* the intercepted representation provides. To do this, we estimate three posteriors using weighted k-nearest neighbors:

$$q(S \mid z'), \qquad q(S \mid y'), \qquad q(S \mid z', y'). \tag{1}$$

Here, $q(S \mid y')$ captures the amount of information about S already exposed by the intended task label (i.e., predicted class). In contrast, $q(S \mid z', y')$ measures what the adversary can infer about S when possessing both the intercepted representation and the task label. The difference between these two quantities represents the ***additional*** semantic vulnerability introduced by the representation.

For a target (z', y'), we define excess leakage as:

$$\text{ExcessKL} = \text{KL}\big(q(S \mid z', y') \parallel q(S \mid y')\big). \tag{2}$$

That is, ExcessKL quantifies how far the attacker's posterior belief shifts when the representation is added on top of the task label. A high ExcessKL indicates that the representation reveals significantly more information about the semantic attribute S than one could guess just by knowing the main task label.

We further decompose the total information leakage into:

$$\text{TotalKL} \approx \text{TaskKL} + \text{ExcessKL} + \varepsilon, \tag{3}$$

where

$$\text{TaskKL} = \text{KL}\big(q(S \mid y') \parallel p(S)\big), \tag{4}$$

and $p(S)$ is the baseline probability of the attribute appearing overall. Intuitively, TaskKL measures how much of the attribute is naturally explained by the task label. In contrast, ExcessKL isolates the unintended leakage coming strictly from the representation. The ε term accounts for estimation noise. Ultimately, *REVEAL* flags attributes as highly exploitable if they provide a large amount of extra information beyond what the task label already reveals. It does this by filtering for attributes where both the absolute ExcessKL and its ratio to TaskKL exceed predefined thresholds.

Intuition. At a high level, we ask a simple question: if an attacker already knows the model's predicted class, how much more can the intercepted representation reveal about a discovered attribute? For example, if the task label is "dog," then some attributes such as "has fur" may already be unsurprising from the class prediction alone. However, if the representation also reveals that the image likely contains a "living room" or a "red leash," then this extra information reflects leakage beyond the intended task. In this sense, $q(S \mid y')$ captures what the attacker could infer from the task output alone, while $q(S \mid z', y')$ captures what the attacker could infer after also observing the representation. Our excess leakage metric measures the gap between these two beliefs, so that larger values indicate that the representation contributes meaningful additional semantic information not explained by the task label itself.

Why KL? We use KL divergence to quantify how much the intercepted representation changes the attacker's posterior belief about an attribute. If observing the representation does not meaningfully change the belief beyond what is already implied by the task label, then the KL value remains small. If the representation substantially sharpens or shifts that belief, then the KL value becomes large. Under this interpretation, ExcessKL captures the additional semantic evidence revealed by the representation.

5 Evaluation

We evaluate *REVEAL* along two research questions. (**RQ1**) Can *REVEAL* surface semantically valid attributes from exposed representations across different model architectures while keeping hallucinated discoveries low? (**RQ2**) How sensitive are these results to key design choices, in particular the neighborhood size k and the choice of LLM?

5.1 Experimental Setup

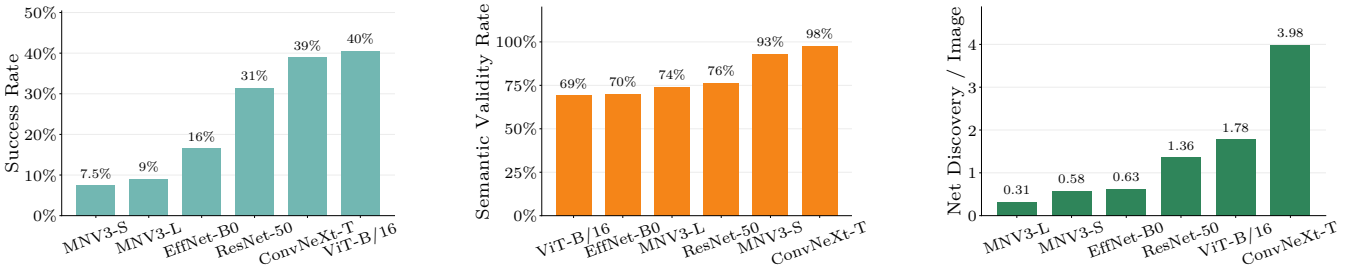
Dataset. We construct a controlled synthetic dataset using Stable Diffusion 3. Each image follows a strict four-part composition: a dominant primary object (the intended task label), a secondary object, a ternary object, and a coherent background scene. These components are explicitly encoded in the text prompts with spatial constraints across diverse settings (e.g., medical, urban, commercial). This setup ensures a controlled separation between the task-explained semantics and the unintended, excess semantic leakage.

Auxiliary Corpus. We evaluate *REVEAL* using the Densely Captioned Images (DCI) dataset [11] as the auxiliary corpus. DCI contains human-authored descriptions, providing rich semantic context for both neighborhood interpretation and weak-label construction.

Models and Representations. We evaluate embeddings produced by multiple ImageNet-1K pre-trained image classification models, including MobileNet variants, ResNet-50, EfficientNet, ConvNeXt-Tiny, and ViT-B, to examine whether the discovered attributes are architecture-specific or broadly present in exposed intermediate representations.

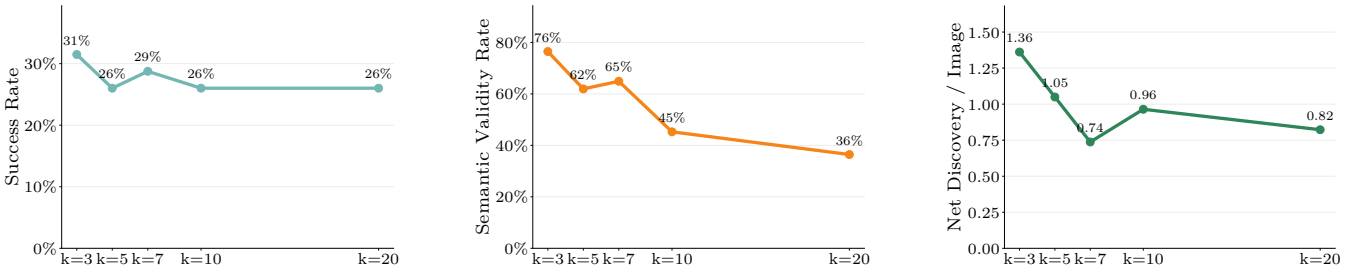
Evaluation Metrics. Because each synthetic image contains one intended task attribute (the primary object) and three non-task attributes, we evaluate the attack using three core metrics where higher is always better. (1) *Success Rate* is the percentage of intercepted embeddings from which the attack successfully infers at least one non-task attribute. (2) *Semantic Validity Rate* is the percentage of discovered attributes that a CLIP-based Vision-Language Model independently verifies as genuinely present in the original image. (3) *Net Discovery per Image* measures the average number of verified semantic attributes per target embedding, minus the number of invalid attributes to penalize LLM hallucination.

Baseline Setting. We establish a standard baseline using our best-performing settings to ensure fair comparisons. This baseline uses a neighborhood size of $k = 3$ and the LLM of GPT-5.4. We apply these settings when comparing different models. When testing the individual impact of either the neighborhood size or the LLM choice, we use a ResNet-50 model and keep all other variables at this standard baseline.



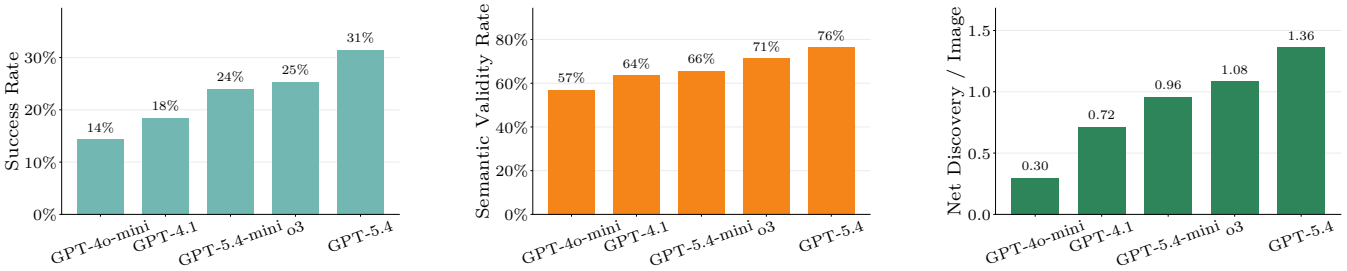
(a) Success Rate **(b) Semantic Valid Rate** **(c) Net Discovery per Image**

Figure 2: Comparison of semantic discovery performance across six deep learning models. From left to right: (2a) Success Rate, (2b) Semantic Valid Rate, and (2c) Net Discovery per Image.



(a) Success Rate **(b) Semantic Valid Rate** **(c) Net Discovery per Image**

Figure 3: Neighborhood-size sensitivity. From left to right: (3a) Success Rate, (3b) Semantic Valid Rate, and (3c) Net Discovery per Image. We use GPT-5.4 and test on ResNet-50. *REVEAL* yields optimal performance at $k = 3$.



(a) Success Rate **(b) Semantic Valid Rate** **(c) Net Discovery per Image**

Figure 4: LLM sensitivity. From left to right: (4a) Success Rate, (4b) Semantic Valid Rate, and (4c) Net Discovery per Image. We set $k = 3$ and test on ResNet-50. *REVEAL* yields optimal performance when GPT-5.4 is used.

5.2 RQ1: Semantic Validity

Figure 2 demonstrates that all tested models leak meaningful non-task information. However, a model's vulnerability depends heavily on its architecture and capacity. Larger models expose significantly more unintended semantics than lightweight models like the MobileNet variants. ConvNeXt-Tiny exhibits the most severe leakage overall. It produces the highest Semantic Validity Rate (97.6%) and the highest Net Discovery per Image (3.980), meaning its exposed attributes are highly accurate and rarely hallucinated. In contrast, while ViT-B/16 exposes unintended attributes more frequently, its leaked representations are noisier and lead to a lower Validity Rate (69.2%) and Net Discovery (1.780). ResNet-50 presents a moderate vulnerability profile, yielding a 31.5% Success Rate, 76.5% Validity, and 1.362 Net Discovery. Interestingly, while MobileNetV3-Small rarely leaks extra information, the few semantic attributes it does expose are highly accurate with a 93.2% Validity Rate.

Takeaway. Semantic leakage is a consistent vulnerability across all tested architectures, with higher-capacity models leaking the most unintended information. Future defenses must therefore actively prevent representations from encoding features outside the primary task.

5.3 RQ2: Sensitivity to Design Choices

We conduct an ablation study to investigate how the attack's configuration, namely neighborhood size and LLM choice, impacts semantic discovery performance.

Impact of Neighborhood Size. Figure 3 demonstrates that smaller neighborhood size consistently yield stronger attack performance. Keeping the LLM constant at GPT-5.4, $k = 3$ achieves the highest metrics across the board (e.g., 31.5% Success Rate, 1.362 Net Discovery). As k increases to 10 or 20, the performance drops sharply. This confirms that overly large neighborhoods dilute the local semantic structures necessary for *REVEAL* to reliably identify unintended attributes.

Impact of LLM Choice. Figure 4 reveals that the reasoning capability of the chosen LLM has a crucial effect on the attack performance. Fixing the neighborhood size at $k = 3$, performance scales almost monotonically with model capability. GPT-5.4 is the optimal choice, followed closely by o3 and GPT-5.4-mini. Conversely, smaller and older models like GPT-4.1 and GPT-4o-mini struggle significantly.

Takeaway. The attack's effectiveness is highly sensitive to both retrieval resolution and reasoning capacity. Small, focused neighborhoods ($k = 3$) are crucial for preventing semantic dilution, and frontier LLMs are required to accurately parse and operationalize the retrieved context.

6 Discussion

Closed-World vs. Open-World Attacks. Prior inference attacks rely on a "checklist" mentality where the adversary guesses which sensitive attributes to target. This gives defenders a false sense of security. They secure systems against a fixed set of attributes but leave unanticipated semantic channels open. *REVEAL* shows that adversaries do not need predefined targets to exploit intermediate representations. Future defenses must restrict the representation's overall capacity to encode non-task features, rather than attempting to suppress a checklist of known attributes.

Incomplete Coverage of the Attack Surface. While *REVEAL* is highly effective, its coverage of the attack surface remains incomplete due to three operational constraints: (1) discovery is bottlenecked by the semantic richness of the attacker's auxiliary corpus; (2) the representation space geometry may not organize certain abstract features into retrievable neighborhoods; and (3) LLM-based summarization may miss non-linguistic features or hallucinate concepts. We therefore do not claim that *REVEAL* discovers all possible semantic leakage. Instead, it provides the first practical capability to discover and exploit unanticipated semantic attributes from intermediate representations.

Limitations and Future Work. Our current evaluation is intentionally controlled, which helps isolate excess semantic leakage from task-relevant information, but it does not fully capture the complexity of real-world images. A natural next step is to validate *REVEAL* on real image datasets with richer visual variation, weaker compositional regularity, and noisier semantic structure. In addition, incorporating human analysis of the discovered attributes would help evaluate whether the inferred concepts are meaningful from a practical perspective and identify which forms of leakage are most sensitive. Finally, because *REVEAL* relies on LLM-based reasoning for attribute discovery and operationalization, its outputs may be sensitive to prompt design, model choice, and inference budget. This sensitivity does not alter the core security finding that open-world leakage exists, but it can affect the consistency of the attack and the reproducibility of the measured results. We therefore treat these factors as practical components of the attack setting and evaluation context.

7 Conclusion

We introduce an open-world inference attack that discovers unintended semantics in intermediate representations, exposing a critical vulnerability in distributed AI systems. By coupling nearest-neighbor retrieval with large language models, our method extracts hidden semantic attributes without prior knowledge. While not exhaustive, it provides the first practical exploit for unanticipated semantic channels. To secure these systems, we encourage future defensive research to move beyond static checklists of known attributes and adopt dynamic, open-world threat models that anticipate the discovery of unknown leakage.

Acknowledgment

This research is partially supported by Samsung Electronics (Grant No. IO240424-09655-01), the National Research Foundation of Korea (NRF), funded by the Ministry of Science and ICT (MSIT) under Grant RS-2024-00464269, the National Key R&D Program of China (2023YFA1009500) and the Fundamental Research Funds for the Central Universities (2025110528-0).

References

[1] G. Alain and Y. Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *Proc. International Conference on Learning Representations (ICLR) Workshop.*

[2] Y. Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics* 48, 1 (2022), 207–219.

[3] M. Bukhari, A. Iqbal, M. N. Aman, and B. Sikdar. 2024. A Representation Learning Induced Property Inference Attack on Machine Learning Models for E-Health. In *Proc. IEEE Global Communications Conference (GLOBECOM).*

[4] Y. Chen, Q. Xu, D. Eliott, Q. Li, and J. Bjerva. 2026. Semantic Leakage from Image Embeddings. *arXiv preprint arXiv:2601.22929* (2026).

[5] M. Coavoux, S. Narayan, and S. B. Cohen. 2018. Privacy-preserving neural representations of text. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP).* 1–10.

[6] Z. He, T. Zhang, and R. B. Lee. 2019. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC).* 148–162.

[7] B. Jayaraman and D. Evans. 2022. Are attribute inference attacks just imputation?. In *Proc. ACM Conference on Computer and Communications Security (CCS).*

[8] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang. 2022. ML-Doctor: Holistic risk assessment of inference attacks against machine learning models. In *Proc. USENIX Security Symposium.*

[9] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *Proc. IEEE Symposium on Security and Privacy (S&P).* 691–706.

[10] Y. Qiu, Y. Liu, H. Yu, H. Fang, B. Chen, S. Xia, and K. Xu. 2025. Revisiting the Privacy Risks of Split Inference: A GAN-based Data Reconstruction Attack via Progressive Feature Optimization. *arXiv preprint* (2025).

[11] J. Urbanek, F. Bordes, P. Astolfi, M. Williamson, V. Sharma, and A. Romero-Soriano. 2024. A Picture is Worth More Than 77 Text Tokens: Evaluating CLIP-Style Models on Dense Captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 26938–26949.

[12] P. Vepakomma, A. Singh, O. Gupta, and R. Raskar. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. In *Proc. ICDM Workshop on Distributed Machine Learning.*