The Label Imitation Game: Turing Test Network for Zero-Shot Pseudo-Label Pruning

Brent A. Griffin[1] and Jason J. Corso[1,2]

[1] Voxel51, {brent,jason}@voxel51.com
[2] University of Michigan

Abstract. Foundation model pseudo-labeling—labeling data strictly via zero-shot inference—enables massive scale, but performance is undermined by hallucinations that evade standard thresholds. To eliminate these errors, we introduce the Turing-inspired **Label Imitation Game (LIG)**, a framework that formalizes pseudo-label pruning as an adversarial interrogation. Rather than filtering labels via isolated thresholds, we use the LIG to train a **Turing Test Network (TTN)**, a task-agnostic "judge" that evaluates candidate pseudo-labels within a dataset-wide context. Experiments across four diverse datasets demonstrate the TTN's robustness, consistently enhancing label accuracy for three state-of-the-art vision-language models without costly supervision or retraining. Crucially, we demonstrate that learned semantic-contextual logic is a robust alternative to spatial-geometric verification, enabling a unique zero-shot task transfer capability—a TTN trained strictly on image classification datasets can effectively prune complex object detection pseudo-labels. This pruning yields F_1-score gains of 28% for the worst-performing baseline categories and 44% with task-specific fine-tuning. Significantly, we also observe **Category Revival**, where the TTN pruning "detoxifies" the training signal for downstream models and enables them to recover from zero recall on transfer-vulnerable classes. The pre-trained TTN models and code are available at https://github.com/voxel51/ttn.

Keywords: Pseudo-labeling · Label Pruning · Zero-Shot Task Transfer · Noise-Robust Learning · Foundation Models · Semi-Supervised Detection

1 Introduction

Can machines annotate data? Foundation vision-language models (VLMs), while powerful for tasks like object detection [11,16,35], often suffer systemic hallucinations when applied to open-world problems [18,32,59]. For pseudo-labeling [29], these hallucinations add training noise that degrades downstream model performance relative to expensive (but accurate) human labeling [37,41]. Accordingly, traditional strategies use fixed thresholds to mitigate noise [15,20,24]. However, our experiments find this often discards authentic, low-confidence labels while retaining structured errors that are "confidently wrong." To address this, there is a critical need for a lightweight, post-hoc "interrogator" that differentiates accurate labels from hallucinations, *ideally* without any costly human supervision or changes to the underlying VLM architecture and downstream training pipeline.

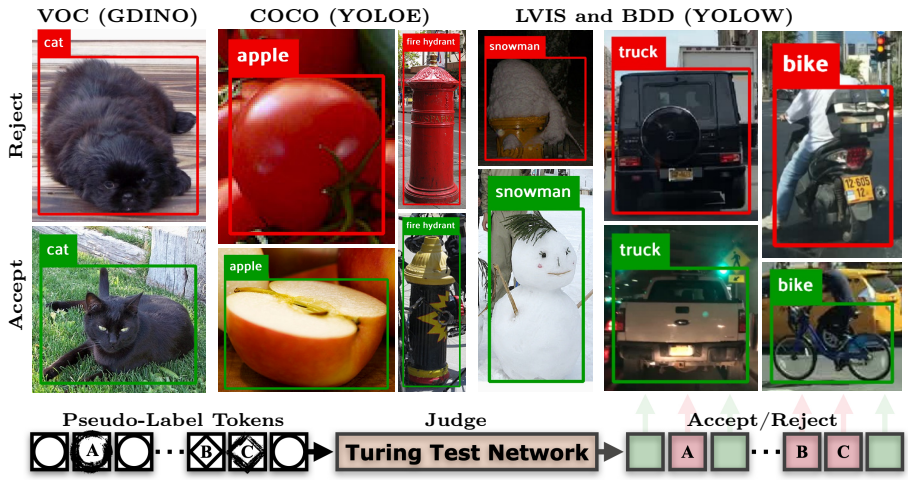

Fig. 1: Zero-Shot Pseudo-Label Pruning. A single Turing Test Network (TTN) trained strictly on image-classification (bottom) finds and rejects systemic VLM hallucinations across diverse detection datasets and pseudo-label architectures (top) while accepting accurate labels (middle). TTN rejects labels for spatial inaccuracy (**A**), semantic inconsistency (**B**), or both (**C**). Visualizations generated using FiftyOne [36].

Inspired by Alan Turing's "Imitation Game" [48], we propose the Label Imitation Game (LIG), which formalizes pseudo-label pruning as an adversarial interrogation where a judge must distinguish between accurate labels and hallucinations. To operationalize this judge, we introduce the Turing Test Network (TTN)—a transformer-based module that rejects erroneous labels by evaluating their contextual plausibility against a dataset-wide reference set (see Fig. 1).

Our primary contributions are: **(i) The Label Imitation Game (LIG):** a novel framework that formalizes label pruning as an adversarial interrogation between a judge and candidate labels (Sec. 4); **(ii) The Turing Test Network (TTN):** a lightweight, transformer-based architecture that *automates* the role of the LIG judge by utilizing non-masked self-attention to evaluate semantic and spatial consistency without class labels (Sec. 5); and **(iii) Zero-Shot Task Transfer:** a cross-task evaluation demonstrating that a TTN trained strictly on image classification can prune object detection pseudo-labels without *any* human input, yielding F_1-score gains of 28% and Category Revival—enabling downstream detectors to recover from zero recall on up to 27 classes (Sec. 6).

2 Related Work

Foundation Model Pseudo Labeling is a scalable alternative to costly manual annotation [15, 37, 41]. After earlier work established the benefits of unified Vision-Language Models (VLMs) [58], grounded language-image pre-training [31] reformulated object detection as a phrase grounding task—aligning text phrases to image regions. This VLM innovation enables massive zero-shot transfer by treating arbitrary detection categories as contextualized text prompts,

which is enhanced in open-vocabulary detectors like Grounding DINO [34] and YOLO-World [6]. These VLMs are powerful tools for "data engines" [1, 52], with innovative processes like using high-confidence predictions in weakly-augmented views to generate pseudo-labels for strongly-augmented ones [47]. Nonetheless, VLMs are fallible and generally limited by a precision-recall trade-off [15].

Hallucinations and Label Noise—predicting entities that are visually absent or contextually inconsistent [18, 32, 59]—are the Achilles heel of VLMs, even occurring at high confidence thresholds [5] (Fig. 2). Thankfully, VLM hallucinations can be treated as a specialized label noise problem [10]. While noise-robust methods commonly use fixed loss thresholds [20, 24], self-adaptive and class-balanced approaches can explicitly identify noisy data [46]. For pseudo-label pruning, specifically, other techniques maximize re-labeling accuracy by selecting subsets that maintain neighborhood prediction confidence [39] or iteratively refining boxes to improve localization [21]. Other spatial-geometric verification methods use embeddings to prune redundant detections [44], but rely on local feature similarity. To expand beyond local heuristics, we leverage a dataset-wide semantic context to selectively prune inaccurate pseudo-labels, addressing global inconsistencies that traditional methods overlook.

Adversarial Interrogation provides a novel framework for pruning. The concept of a Visual Turing Test was first proposed to evaluate scene interpretation through binary interrogation [13], while later adversarial variants for dialogue use a "Judge" to discriminate between human and machine responses [12]. This draws a direct analogy to generative adversarial networks [14], which popularized the minimax game between a generator and discriminator. This emerging "LLM-as-a-Judge" paradigm provides a scalable alternative to human evaluation by using foundation model reasoning to assess output quality across diverse tasks [30, 57]. As an alternative to high-cost LLMs, adversarial filters use an iterative greedy algorithm to remove classification data biases [4], but this approach precludes reasoning for complex spatial-semantic interrogation. We address this with the Label Imitation Game (LIG), a general framework that formalizes label pruning as an adversarial interrogation independent of source models. By decoupling evaluation from specific task heads, the LIG enables our lightweight Turing Test Network to act as a task-agnostic judge that interrogates the semantic plausibility of candidate pseudo-labels within a global reference context.

3 Problem Formulation: The Label Noise Challenge

Assume a model generates pseudo-labels $\hat{\mathbb{Y}}$ on an unlabeled dataset $\mathbb{D}_{\text{U}} = \{\mathbf{x}_i\}_{i=1}^{M}$. Here, data examples $\mathbf{x}_i$ are drawn i.i.d. from an underlying distribution P, and $\hat{\mathbb{Y}} = \{\hat{\boldsymbol{y}}_i\}_{i=1}^{M}$ comprises variable-length label sets ($|\hat{\boldsymbol{y}}_i| \geq 0$) for each $\mathbf{x}_i$.

In practice, pseudo-label accuracy is highly variable and, even with high-confidence thresholds, models often produce hallucinations (see Fig. 2). Our goal is to (i) selectively prune these inaccurate labels to increase precision, while (ii) preserving accurate labels to maintain recall. Accordingly, we evaluate and prune $\hat{\mathbb{Y}}$ to select label subset $\tilde{\mathbb{Y}} = \{\tilde{\boldsymbol{y}}_i\}_{i=1}^{M}$ and formulate the **pseudo-label**

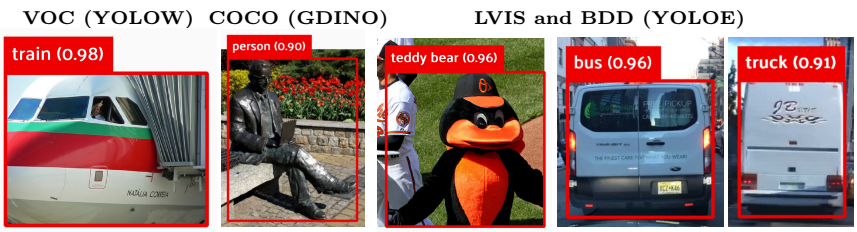

Fig. 2: TTN Zero-Shot Rejections on High-Confidence Pseudo-Labels

pruning problem as

$$\arg\max_{\tilde{\mathbb{Y}}\subseteq\hat{\mathbb{Y}}} \mathbb{E}_{\mathbf{x},\boldsymbol{y}\sim P}[F_1(\boldsymbol{y};\tilde{\boldsymbol{y}})], \tag{1}$$

where $\boldsymbol{y}$ is the ground truth and the F_1 score is the harmonic mean of precision and recall. Specifically, $F_1 = 2\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \in [0,1]$, precision $= \frac{\text{TP}}{\text{TP}+\text{FP}} \in [0,1]$, recall $= \frac{\text{TP}}{\text{TP}+\text{FN}} \in [0,1]$, true positives (TP) is the number of accurate pseudo-labels, and false negatives (FN) is the number of $\boldsymbol{y}$ labels without a corresponding TP. In plain words, **precision** is the frequency of pseudo-labels being correct, **recall** is the frequency of ground-truth instances being correctly pseudo-labeled, and the goal of Eq. (1) is maximizing the F_1 **score**, which emphasizes label performance across *both* metrics ($F_1 = 1$ for perfect precision and recall).

Pseudo-labeled datasets are commonly used for **downstream model training**. However, if left unpruned, pseudo-labels corresponding to hallucinations will induce **inductive interference** that misleads downstream model convergence. Thus, we combine pruned pseudo-labels $\tilde{\mathbb{Y}}$ with $\mathbb{D}_{\text{U}}$ to generate the "detoxified" dataset $\tilde{\mathbb{D}} = \{(\mathbf{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{M}$. We evaluate $\tilde{\mathbb{D}}$ by reformulating pruning from Eq. (1) explicitly for downstream model training as

$$\arg\min_{\tilde{\mathbb{Y}}\subseteq\hat{\mathbb{Y}}} \mathbb{E}_{\mathbf{x},\boldsymbol{y}\sim P}[\ell(\mathbf{x},\boldsymbol{y};f_{(\tilde{\mathbb{D}})})], \tag{2}$$

where ℓ is the task-specific loss function and $f_{(\tilde{\mathbb{D}})}$ is a model trained on $\tilde{\mathbb{D}}$. In plain words, the goal of Eq. (2) is to prune pseudo-labels ($\tilde{\mathbb{Y}} \subseteq \hat{\mathbb{Y}}$) to train a downstream model (f) that most accurately predicts ground-truth labels ($\boldsymbol{y}$) for unseen images ($\mathbf{x}$) drawn from the underlying data distribution (P).

4 Adversarial Interrogation via the Label Imitation Game

The Turing test evaluates a machine's ability to exhibit intelligent behavior indistinguishable from that of a human [48]. Likewise, our goal is to evaluate which pseudo-labels are indistinguishable from those of a human annotator, implicitly denoting high accuracy. We formalize this as the **Label Imitation Game** (LIG).

We formulate label tests in the context of a game involving a set of **Ideal** and **Candidate** labels and a **Judge**. The Judge is given all labels and must determine which belong to each set. We use LIG to train the Turing Test Network, which later serves as the Judge at inference to reject and prune pseudo-labels, solving $\tilde{\mathbb{Y}} \subseteq \hat{\mathbb{Y}}$ in Eqs. (1) & (2). We use a tokenized representation (Sec. 4.1) for the LIG (Sec. 4.2), but these games are general to the original labels and data.

4.1 Tokenizing Labels & Data

The transformer architecture (the *de facto* standard for NLP [49]) operates on a tokenized representation of data to iteratively compare context across all inputs before determining final outputs. To utilize this iterative comparison capability for the LIG, we fuse labels $\boldsymbol{y}$ and data $\mathbf{x}$ into tokens, specifically

$$g(\mathbf{x}, \boldsymbol{y}) = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n\}, \tag{3}$$

where g is a function operating on $\mathbf{x}$ and $\boldsymbol{y}$, $n = |\boldsymbol{y}| = |g(\mathbf{x}, \boldsymbol{y})|$, and each output token $\boldsymbol{g}_i$ is a lower-dimension representation than $\mathbf{x}$ (we specify g in Sec. 5). Extending Eq. (3), we can generate arbitrary tokens across m data examples as

$$\mathbb{G} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N\} = \{g(\mathbf{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}. \tag{4}$$

Using Eq. (4), we reformulate label pruning as **token pruning**, i.e., selecting $\tilde{\mathbb{G}} \subseteq \hat{\mathbb{G}}$, where $\hat{\mathbb{G}} = \{\hat{\boldsymbol{g}}_i\}_{i=1}^{\hat{N}} = \{g(\mathbf{x}_i, \hat{\boldsymbol{y}}_i)\}_{i=1}^{M}$ and $\hat{N} = |\hat{\mathbb{Y}}|$. Furthermore, because there is a one-to-one correspondence between tokens and labels, tokens can map directly back to labels as $g^{-1} : \tilde{\mathbb{G}} \to \tilde{\mathbb{Y}}$ in Eq. (1) for bijective equivalence.

4.2 Label Imitation Games

For **Game 1**, we use Eq. (4) to generate a set of tokens $\mathbb{G}^* = \{\boldsymbol{g}_i^*\}_{i=1}^{N^*}$ from accurately labeled data $\{\mathbf{x}_i, \boldsymbol{y}_i^*\}_{i=1}^{m^*}$ and $\hat{\mathbb{G}} = \{\hat{\boldsymbol{g}}_i\}_{i=1}^{\hat{N}}$ from pseudo-labeled data $\{\mathbf{x}_i, \hat{\boldsymbol{y}}_i\}_{i=1}^{m}$. The **Judge's** objective is to predict which tokens in the combined set correspond to pseudo-labels. We formulate this **Pseudo-Label Game** as

Game 1 (PLG) *Given* $\mathbb{G}^* \cup \hat{\mathbb{G}}$*, identify which tokens belong to* $\hat{\mathbb{G}}$*.*

The intuition of PLG is if a **Candidate** token $\hat{\boldsymbol{g}}_i \in \hat{\mathbb{G}}$ is distinguishable from the **Ideal** set $\mathbb{G}^*$, its corresponding pseudo-label is likely inaccurate and should be rejected. Conversely, as $\hat{\boldsymbol{g}}_i$ approaches $\mathbb{G}^*$ accuracy, it becomes effectively indistinguishable. In fact, experiments show that carefully chosen $\hat{\boldsymbol{g}}_i$ can even serve as Ideal tokens $\mathbb{G}^*$ (we call this "Self-Referential Pruning" in Sec. 5.2).

For **Game 2**, assume each token is associated with a latent category $c_i \in \mathcal{C}$ from its underlying label and data. Extending Eq. (4), we denote $c_i \; \forall \; \boldsymbol{g}_i$ as

$$\mathbb{G} = \{\boldsymbol{g}_1^{(c_1)}, \boldsymbol{g}_2^{(c_2)}, \ldots, \boldsymbol{g}_N^{(c_N)}\} = \{g(\mathbf{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}. \tag{5}$$

In this game, by construction, the **Ideal** tokens $\mathbb{G}^{(c^*)}$ are homogeneous w.r.t. a target category c^*, s.t. $c_i = c^*, \forall \; \boldsymbol{g}_i^{(c_i)} \in \mathbb{G}^{(c^*)}$. Conversely, **Candidate** tokens $\mathbb{G}^{(\neg c^*)}$ are strictly disjoint from c^*, s.t. $c_i \neq c^*, \forall \; \boldsymbol{g}_i^{(c_i)} \in \mathbb{G}^{(\neg c^*)}$. The **Judge's** objective is to predict which tokens in the combined set belong to disjoint categories $c_i \neq c^*$. We formulate this **Out-of-Category Game** as

Game 2 (OCG) *Given* $\mathbb{G}^{(c^*)} \cup \mathbb{G}^{(\neg c^*)}$*, identify which tokens belong to* $\mathbb{G}^{(\neg c^*)}$*.*

Fig. 3: TTN Zero-Shot Rejections on Poorly-Localized Pseudo-Labels

Notably, the OCG Judge is not informed of the target category c^*. To ensure the objective is not underdetermined, we implement OCG s.t. $|\mathbb{G}^{(c^*)}| > |\mathbb{G}^{(\neg c^*)}|$. This constraint provides sufficient referential density for the Judge to infer target identity, resolve categorical ambiguity, and distinguish the Ideal tokens from Candidates. The intuition of OCG is if a token $\boldsymbol{g}_i \in \mathbb{G}^{(\neg c^*)}$ is distinguishable from the ideal set $\mathbb{G}^{(c^*)}$, its corresponding pseudo-label is semantically inconsistent or noisy and should be rejected. A substantial benefit of OCG is generating numerous training examples for the Turing Test Network (details in Sec. 5).

Finally, denoting PLG 1 and or OCG 2 as h, we find $\tilde{\mathbb{Y}} \subseteq \hat{\mathbb{Y}}$ in Eq. (1) as

$$\tilde{\mathbb{Y}} = g^{-1}(\tilde{\mathbb{G}}) = g^{-1}\big(h(\mathbb{G}^* \cup \hat{\mathbb{G}}) \cap \hat{\mathbb{G}}\big), \tag{6}$$

where $\tilde{\mathbb{G}}$ are Candidate pseudo-label tokens $\hat{\boldsymbol{g}}_i \in \hat{\mathbb{G}}$ accepted by h when provided $\mathbb{G}^* \cup \hat{\mathbb{G}}$ and g^{-1} maps the tokens back to their original labels (Eq. (4)).

5 Learning to Judge: The Turing Test Network

We introduce the **Turing Test Network** (TTN) to *automate* the role of LIG Judge. Specifically, we employ a transformer-based discriminator trained on a LIG-based binary cross-entropy loss to identify label errors (Sec. 5.1). We train the baseline TTN strictly on classification data but in a class-agnostic manner, withholding categorical labels across 1,303 classes from eight datasets to ensure the TTN learns latent semantic relationships rather than basic class mapping. After training, the TTN exhibits an emergent capacity to build and evaluate semantic (Fig. 2) and spatial context (Fig. 3) solely from input sequences, enabling zero-shot transfer to object detection without supervision (Sec. 5.2). Finally, we introduce a task-specific variant (TTN_{D}) fine-tuned on detection data, pseudo-labels, and ≤ 100 human labels per class for enhanced performance (Sec. 5.3).

Network Architecture: TTN's architecture consists of (i) a frozen CLIP [43] feature extractor that encodes label-defined image patches as $\boldsymbol{f}_i \in \mathbb{R}^{768}$ and (ii) a custom-learned tokenizer $\boldsymbol{g}_i(\boldsymbol{f}_i) \in \mathbb{R}^{512}$ that enables the (iii) transformer-based reasoning block to evaluate each token and output Accept/Reject logits z_i (Fig. 4). For full details, please refer to Sec. S1 in the Supplementary Material.

5.1 Training via the Label Imitation Game

We train the TTN by minimizing a weighted binary cross-entropy (BCE) loss over LIG outcomes. For a sequence of $n = |\mathbb{G}|$ tokens, the objective function is

$$\mathcal{L}_{\text{BCE}}(\boldsymbol{w}) := -\sum_{i=1}^{n} \Big[p_w y_i \cdot \log\big(\sigma(z_i)\big) + (1 - y_i) \cdot \log\big(1 - \sigma(z_i)\big) \Big], \tag{7}$$

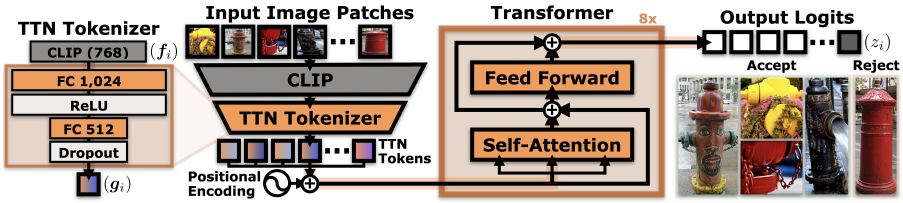

Fig. 4: Turing Test Network (TTN) Architecture. The TTN processes image patches via a TTN Tokenizer and an eight-layer Transformer. Using non-masked self-attention, TTN evaluates relational and sequential context to produce Accept/Reject logits (z_i) for each input without any class labels, enabling zero-shot task transfer.

where $\boldsymbol{w}$ are trainable parameters of the tokenizer and transformer, $y_i \in \{0, 1\}$ is the ground-truth indicator assigning $y_i = 0$ to the **Accept** set and $y_i = 1$ to the **Reject** set, p_w is a positive weight applied to Reject training examples (conceptually, pruning aggression), z_i is the output logit per token $\boldsymbol{g}_i \in \mathbb{G}$, and $\sigma(z_i) = \frac{1}{1+e^{-z_i}}$ is the model's estimated probability that $\boldsymbol{g}_i$ should be rejected.

To ensure the TTN generalizes across diverse failure modes, we construct training batches by sampling tokens $\boldsymbol{g}_i$ from a categorical **training distribution** $P(\mathbb{S})$ over the following four mutually exclusive states $\mathbb{S} \in \{1, 2, 3, 4\}$:

$\mathbb{S} = 1$ (Ideal): High-fidelity human-label token with target category c^*.
$\mathbb{S} = 2$ (OCG-Human): Human-label token with disjoint category $c_i \neq c^*$.
$\mathbb{S} = 3$ (OCG-Pseudo): Pseudo-label token with disjoint category $c_i \neq c^*$.
$\mathbb{S} = 4$ (PLG): Pseudo-label token of any fidelity and target category c^*.

By construction, $y_i = 1$ if $\mathbb{S} = 1$ and $y_i = 0$ otherwise in Eq. (7). We denote the sampling probability for each state as $\pi_k = P(\mathbb{S} = k)$, where $\sum \pi_k = 1$. This formulation enables control of LIG difficulty by adjusting the training failure distribution. For OCG ($\mathbb{S} = \{2, 3\}$), all disjoint categories $c_i \neq c^*$ have an equal chance of being selected, but we only use one disjoint category per game.

Training Protocol: Each training sequence includes references for context prior to candidates. Specifically, we assign the first 10 positions sampled exclusively from the Ideal state ($\mathbb{S} = 1$) to the **Reference Set**. To ensure a robust reference context, we limit Reference Sets to only include categories with $N > 10$ instances. The final position is reserved for the **Candidate Slot**, which is sampled from the full distribution $P(\mathbb{S})$. This forces the TTN to utilize the Reference Set's context to interrogate the Candidate's fidelity. While experiments indicate the TTN can handle an arbitrary number of Reference and Candidate Slots, we utilize this 10:1 ratio for all training results. Notably, a disjoint category may still serve as a Reject Candidate ($\mathbb{S} \neq 1$) even if it fails to meet the Reference Set threshold (i.e., $N \leq 10$ instances), provided it contains at least one representative label.

Optimization: We optimize the baseline TTN using $p_w = 1$ in Eq. (7), a batch size of 256, and an SGD optimizer (5×10^{-4} weight decay, 5×10^{-3} learning rate, implemented via PyTorch [40]). Each distinct dataset contributes 1,000 training sequences per epoch, with a 20% held-out validation set for model selection. To ensure the TTN is a balanced and domain-agnostic judge, we se-

Table 1: TTN Classification LIG Evaluation. TTN trains on 80% of images from all categories and is evaluated on the remaining 20% for validation. The Accept/Reject validation accuracies are for Ideal ($\mathbb{S} = 1$) and OCG ($\mathbb{S} = 2$) Candidates, respectively.

Dataset	Number of Classes	Number of Images	Image Size	TTN 80/20 Split Acc. Accept	Reject	Overall
ImageNet [8]	1,000	1,281,167	Varied	97.7%	94.8%	96.2%
Food-101 [3]	101	101,000	$\leq$512	97.7%	95.3%	96.5%
CIFAR100 [28]	100	50,000	32×32	77.6%	85.4%	81.5%
MIT Indoor Scenes [42]	67	15,620	$\geq$200	98.9%	96.5%	97.7%
Fashion-MNIST [53]	10	60,000	28×28	97.2%	92.5%	94.8%
CIFAR10 [28]	10	50,000	32×32	98.3%	97.8%	98.0%
EuroSAT [22]	10	27,000	64×64	98.6%	97.6%	98.1%
VTID2 [2]	5	4,356	Varied	100.0%	99.8%	99.9%

lect the final parameters $\boldsymbol{w}$ from the checkpoint achieving the highest validation accuracy. Specifically, we evenly weight the Accept ($\mathbb{S} = 1$) and Reject ($\mathbb{S} \neq 1$) accuracies per dataset, then average across all datasets in Tab. 1. This dual-averaging strategy prevents the TTN from over-fitting to specific image domains or majority-class distributions. Notably, rotating disjoint categories $c_i \neq c^*$ evenly within each dataset also reduces memorizing specific features for the 1,303 overall classes. The TTN trains for 100K epochs (hold-out selection at the 90K checkpoint) in 61.05 hours on a single NVIDIA L40S GPU. This modest computational requirement—relative to the foundation models it evaluates—highlights TTN efficiency as a general lightweight pruning module.

5.2 Zero-Shot Cross-Task Transfer

TTN's zero-shot cross-task transfer performance—pruning object detection labels with a model trained purely on image classification (Figs. 1 to 4)—stems directly from our training protocol, which forces the TTN to identify robust semantic boundaries purely from diverse, high-fidelity input sequences (Tab. 1). Specifically, the TTN training sampling distribution $P(\mathbb{S})$ uses only human-verified sources, $\pi_1 = \pi_2 = \frac{1}{2}$, with zero pseudo-label probabilities, $\pi_3 = \pi_4 = 0$. By excluding noisy pseudo-labels and detection-specific formats during this phase, we ensure the TTN internalizes a robust intrinsic semantic logic. This foundation generalizes to novel tasks and categories in a zero-shot manner, requiring no task-specific supervision. Furthermore, including low-resolution datasets (e.g., CIFAR, Fashion-MNIST) acts as an inherent curriculum for scale variation.

TTN LIG Evaluation: As detailed in Tab. 1, the TTN achieves high validation accuracy ($> 94\%$ Overall on 7/8 datasets) across diverse classification domains. Notably, the model maintains robust performance even on low-resolution data (e.g., 98.0% on CIFAR10), which is representative of the visual degradations typical of tiny object pseudo-labels. By learning to judge pixelated and high-fidelity labels for numerous classes, the TTN develops a domain-agnostic semantic logic that facilitates effective zero-shot transfer from classification to detection.

Self-Referential Pruning: After foundation model pseudo-labeling, the TTN prunes detection labels *without* human supervision by constructing a dataset-spanning "Ideal" set from the 1,000 highest-confidence pseudo-labels per target

Table 2: TTN_D Detection LIG Evaluation. We train TTN_D with VLM pseudo-labels and ≤ 100 human labels per class. Accept/Reject validation accuracies are for $\mathbb{S} = 1/\mathbb{S} = \{2, 3, 4\}$ Candidates, respectively. TTN_D uses a lower p_w (Eq. (7)) to prioritize Accept accuracy; Reject accuracy varies with instance density and class complexity.

Dataset	Number of Training Classes	Images	Objects	Objects per Image	TTN_D 80/20 Split Acc. Accept	Reject	Overall
VOC [9]	20	16,551	40,058	2.42	95.1%	84.2%	89.7%
COCO [33]	80	118,287	849,945	7.19	96.8%	59.2%	78.0%
LVIS [19]	1,203	100,170	1,270,141	12.68	95.8%	50.6%	73.2%
BDD [56]	10	70,000	1,286,871	18.38	93.0%	63.0%	78.0%

category c^*, $|\mathbb{G}^{(c^*)}| \leq 1,000$. We then perform a Self-Referential LIG for each category, where the TTN evaluates *every* pseudo-label candidate $\hat{\boldsymbol{g}}_i^{(c^*)} \in \hat{\mathbb{G}}^{(c^*)}$ (including "Ideal" selections) against 10 reference samples drawn from $\mathbb{G}^{(c^*)}$. We aggregate the Accept/Reject logits for each candidate (z_i) across multiple games to cover all available references and increase robustness; if a candidate's aggregate score is > 0, it is pruned (Eq. (6)). This iterative process allows the TTN to identify and reject systemic hallucinations, even if they had high initial confidence (Fig. 2). Categories lacking sufficient reference labels ($|\mathbb{G}^{(c^*)}| < 10$) are preserved with their original pseudo-labels to prevent uncontextualized pruning.

5.3 Task-Specific Fine-Tuning

To maximize detection pruning performance, we test task-specific fine-tuning (TTN_D) with the full LIG training sampling distribution $P(\mathbb{S})$. From TTN weights, we fine-tune a single TTN_D model on all detection datasets in Tab. 2 simultaneously, spanning web images to autonomous driving. This multi-domain approach ensures TTN_D is a robust generalist across diverse visual contexts while mastering the specific spatial and semantic nuances of the object detection task.

For the Ideal set ($\mathbb{S} = 1$), TTN_D uses up to 100 human-verified detection labels for each target category (subject to availability), which are split 80/20 for the train/validation protocol. For label efficiency, the 80 training labels are reused as the Ideal set for pseudo-label pruning at inference ($|\mathbb{G}^{(c^*)}| \leq 80$).

To generate the remaining training distributions ($\mathbb{S} = \{2, 3, 4\}$), TTN_D uses three Vision-Language Models (VLMs): YOLO-World (YOLOW) [6], YOLOE [50], and Grounding DINO-Tiny (GDINO) [34], each sampled at a confidence threshold of $\tau = 0.1$. To maximize the hallucination signal relative to the Ideal set, we limit training examples from each VLM to the lowest-confidence quartile of pseudo-labels per class. Each VLM-dataset combination contributes 1,000 training sequences per epoch, exposing TTN_D to the diverse error characteristics and spatial inaccuracies of multiple VLM architectures.

TTN_D adopts the baseline TTN training configuration with three modifications: (i) a non-uniform sampling distribution of $\pi_1 = \frac{1}{2}$, $\pi_2 = \frac{1}{4}$, $\pi_3 = \pi_4 = \frac{1}{8}$; (ii) a reduced pruning aggression ($p_w = 0.2$ in Eq. (7)) to instill a conservative bias that minimizes false rejections, ensuring the TTN_D only prunes when there is strong evidence of a hallucination; and (iii) a 0.02 learning rate for 4K epochs (with selection at the 2K checkpoint), requiring only 4.28 hours of fine-tuning.

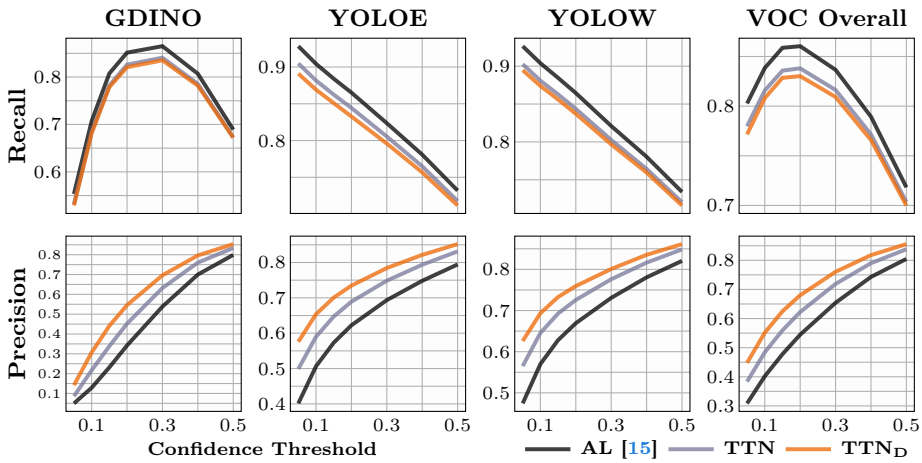

Fig. 5: Recall and Precision across VLM Models and Confidence Thresholds on VOC. YOLOW, YOLOE have higher recall and precision at low thresholds values, likely due to better internal non-maximum suppression of predicted labels. TTN prunes away predicted label failures, which incidentally decreases recall but more substantially increases precision. Additional metric plots across all datasets provided in Fig. S3.

After fine-tuning, TTN_D achieves high validation accuracies (73–90%, Tab. 2). Accept accuracy (Candidate $\mathbb{S} = 1$) consistently exceeds 93% across all domains. Reject accuracy ($\mathbb{S} = \{2, 3, 4\}$) naturally reflects the varying instance density and class complexity of the target datasets, which we explore further in Sec. 6.

6 Experimental Evaluation of the LIG and TTN

6.1 Experimental Setup

Our experimental setup is motivated by the comprehensive **Auto-Labeling** (AL) benchmark [15]. This foundation model pseudo-labeling benchmark uses GDINO, YOLOE, and YOLOW VLMs across various confidence thresholds τ and four diverse detection datasets: VOC, COCO, LVIS, and BDD. We use $\tau \in [0.05, 0.5]$ to improve peak AL F_1-score performance via TTN pruning ($\tilde{\mathbb{Y}} \subseteq \hat{\mathbb{Y}}$ in Eq. (1)). The AL benchmark includes training downstream detectors strictly on pseudo-labels, which provides another evaluation for TTN performance gains ($\tilde{\mathbb{D}}$ in Eq. (2)). In addition to TTN, we include **LIG*** as an Oracle judge that prunes all candidate pseudo-labels with IoU < 0.5. This represents an IoU-specific upper bound for LIG-based pruning, though it preserves spatial label redundancies. See **Supplementary Material for method comparisons, ablations, runtime, extended benchmark results,** and full experiment details (Secs. S2 to S6).

6.2 Interrogating the Signal: Pseudo-Label Evaluation

We first evaluate the effectiveness of the TTN as a standalone pseudo-label filter. As shown on VOC in Fig. 5, TTN consistently improves precision across

Table 3: Combined Label and Downstream Model Evaluation. Results are averaged over VLM model, confidence threshold, and class, except for Labels, which is the total number for all classes. Overall is additionally averaged over all datasets.

Dataset	Method	Mean of VLM, Confidence Threshold, & Class						
		Labels	**Recall**	**Prec.**	F_1	F_2	**mAP$_{50}$**	**mAP$_{50:95}$**
VOC	LIG*	39,008	0.816	0.914	0.845	0.825	0.707	0.500
	TTN$_{\text{D}}$	61,337	0.788	**0.677**	**0.695**	**0.734**	**0.685**	**0.486**
	TTN	78,403	0.794	0.628	0.662	0.716	0.675	0.478
	AL [15]	134,899	**0.815**	0.562	0.615	0.690	0.656	0.466
COCO	LIG*	627,792	0.594	0.908	0.689	0.626	0.454	0.315
	TTN$_{\text{D}}$	1,565,002	0.582	0.537	0.511	**0.534**	**0.436**	0.302
	TTN	1,192,383	0.538	**0.571**	**0.513**	0.515	0.434	**0.303**
	AL	1,834,962	**0.593**	0.514	0.499	0.531	0.432	0.299
LVIS	LIG*	290,613	0.137	0.600	0.191	0.154	0.059	0.041
	TTN$_{\text{D}}$	1,659,673	0.136	**0.116**	**0.079**	**0.087**	**0.060**	**0.042**
	TTN	1,508,817	0.131	**0.116**	**0.079**	0.086	0.059	**0.042**
	AL	1,777,051	**0.137**	0.112	0.078	**0.087**	0.059	**0.042**
BDD	LIG*	426,726	0.267	0.837	0.376	0.301	0.306	0.178
	TTN$_{\text{D}}$	773,755	0.243	**0.459**	**0.262**	0.235	**0.269**	**0.156**
	TTN	854,393	0.242	0.428	0.249	0.227	0.252	0.148
	AL	1,084,056	**0.267**	0.411	0.256	**0.241**	0.253	0.147
Overall	LIG*	346,035	0.454	0.815	0.525	0.476	0.382	0.259
	TTN$_{\text{D}}$	1,014,942	0.437	**0.447**	**0.387**	**0.398**	**0.362**	**0.247**
	TTN	908,499	0.426	0.436	0.376	0.386	0.355	0.243
	AL	1,207,742	**0.453**	0.400	0.362	0.387	0.350	0.238

all VLM models and confidence thresholds. While pruning naturally results in an incidental decrease in recall, the more substantial precision increases lead to overall F_1-score improvements across all datasets (Tab. 3). Specifically, TTN and TTN$_{\text{D}}$ improve Overall F_1 by 4% and 7%, respectively, advancing toward the 45% gain set by the LIG* Oracle. Notably, LIG* exhibits an epsilon improvement in recall due to matching disambiguation. Essentially, by rejecting high-entropy labels, we reduce Hungarian matching interference and enable higher-fidelity pairings between the accepted pseudo-labels and ground truth during evaluation.

We also evaluate the F_2 score, which favors recall $\left(\frac{5\text{TP}}{5\text{TP}+4\text{FN}+\text{FP}} \in [0,1]\right)$, to measure the trade-off between pruning and signal preservation. TTN$_{\text{D}}$ improves Overall F_2 by 3%, demonstrating that its task-specific logic "detoxifies" the label set while preserving high-recall candidates for downstream training.

6.3 Detector Training and Evaluation on Detoxified Labels

We further evaluate the utility of our pruning framework by training downstream object detectors on the detoxified pseudo-labeled datasets. Specifically, for each dataset, we train a YOLO11n detector [26] on unlabeled training images paired with TTN-pruned pseudo-labels ($\tilde{\mathbb{D}}$ in Eq. (2)). After training, these detectors are then evaluated on their respective human-labeled validation sets. Critically, these validation sets (both images and labels) are entirely disjoint and unseen by both the TTN and the downstream detectors, ensuring a robust measure of how TTN pruning improves generalization from VLM supervision.

As shown in Tab. 3, training on TTN-pruned datasets results in consistent mAP$_{50}$ and mAP$_{50:95}$ improvements across all VLM architectures and datasets.

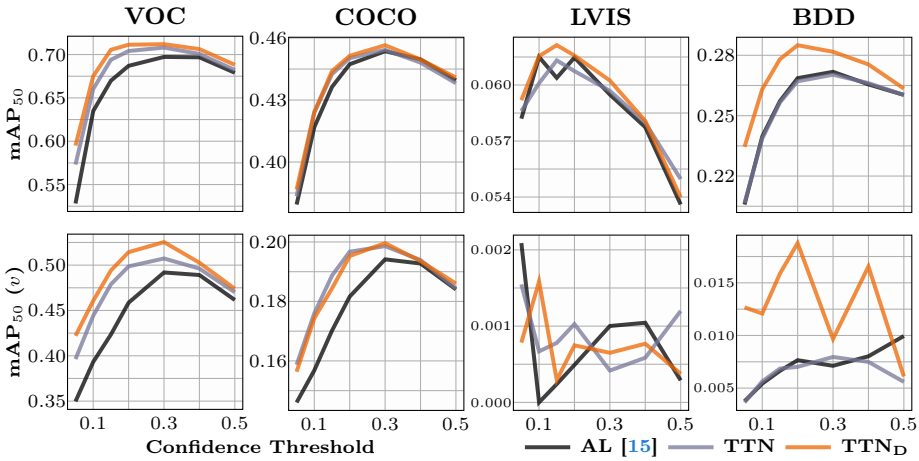

Fig. 6: Downstream Model Evaluation across all Confidence Thresholds and Datasets. Results are average over all VLM models (see Fig. 5 right). Row two results evaluate the vulnerable classes (v, Sec. 6.4). $\text{mAP}_{50:95}$ plots are provided in Sec. S5.

Remarkably, although TTN_D does not reject nearly as many pseudo-labels as the LIG* Oracle (see Labels in Tab. 3), TTN_D closes 38% of the mAP_{50} and 43% of the $\text{mAP}_{50:95}$ gap between AL and LIG* performance. This indicates that TTN_D successfully rejects many of the VLM hallucinations that are particularly disruptive for downstream training. As illustrated in Fig. 6, mAP_{50} gains are particularly pronounced at lower confidence thresholds (τ). By detoxifying the low-τ, high-recall regime, the TTN effectively expands the usable training distribution, allowing the downstream detector to learn from a larger, yet cleaner, set of examples. With TTN's generalization across categories established, we focus on the most challenging, transfer-vulnerable classes in Sec. 6.4.

6.4 Category Revival: Rescuing Vulnerable Classes

To understand TTN performance on challenge categories, we perform a targeted evaluation for AL's [15] maximum and non-zero minimum F_1-score classes in Tab. 4. While TTN F_1 gains are consistent across VLM models and datasets for the Max. F_1 classes, gains for the challenge classes are particularly significant (e.g., $> 300\%$ for GDINO Chair). For the Min. F_1 class mean, specifically, **zero-shot TTN increases F_1 by 28% and TTN_D increases F_1 by 44%**, effectively "raising the floor" for VLM-driven labeling. Remarkably, LIG* Oracle pruning increases Min. F_1 by 432%, indicating systemic VLM hallucinations that would have induced severe inductive interference in downstream models.

Motivated by this finding, we perform a broader analysis of these challenge categories in downstream model training. Specifically, we evaluate transfer-**vulnerable classes** (v), defined per-VLM as the bottom quartile of classes ranked by AL's [15] downstream mAP_{50} at $\tau = 0.1$. As shown in Fig. 6, TTN pruning consistently improves mAP_{50} across confidence thresholds and datasets

Table 4: F_1 Score on Max. & Min. (Non-Zero) AL Classes at $\tau = 0.1$. Relative to AL [15], TTN and TTN_D increase F_1 by 4% and 7% respectively for the best performing classes (top) and 28% and 44% for the worst performing classes (bottom).

Method	VOC F_1 GDINO	VOC F_1 YOLOE	VOC F_1 YOLOW	COCO F_1 GDINO	COCO F_1 YOLOE	COCO F_1 YOLOW	BDD F_1 GDINO	BDD F_1 YOLOE	BDD F_1 YOLOW	Mean
Max. F_1	**Cat**	**Cat**	**Cat**	**Cat**	**Bear**	**Giraffe**	**Car**	**Car**	**Person**	
LIG*	0.747	0.940	0.948	0.675	0.892	0.935	0.693	0.577	0.656	0.785
TTN_D	**0.687**	**0.907**	**0.915**	**0.602**	0.819	0.873	**0.478**	0.519	**0.545**	**0.705**
TTN	0.649	0.893	0.903	0.579	**0.848**	**0.878**	0.443	0.479	0.518	0.688
AL [15]	0.585	0.822	0.851	0.528	0.784	0.861	0.438	**0.535**	0.543	0.661
Min. F_1	**Chair**	**Chair**	**Sofa**	**Handbag**	**Toaster**	**Toaster**	**Train**	**Motor**	**Train**	
LIG*	0.641	0.892	0.940	0.516	0.840	0.686	0.323	0.018	0.319	0.575
TTN_D	**0.204**	**0.520**	**0.436**	0.047	**0.194**	**0.186**	**0.005**	**0.016**	**0.116**	**0.192**
TTN	0.172	0.489	0.391	**0.051**	0.180	0.175	0.002	0.014	0.058	0.170
AL	0.054	0.364	0.319	0.045	0.174	0.166	0.002	**0.016**	0.058	0.133

Table 5: Vulnerable-Class (v) Performance Evaluation. We evaluate pseudo-label quality and downstream detector performance on the transfer-vulnerable classes (bottom quartile of AL [15] mAP_{50}). TTN and TTN_D consistently outperform AL, mitigating the inductive interference caused by vision-language model hallucinations.

Dataset	Method	Mean of VLM, Confidence Threshold, & Class Labels	Recall	Prec.	F_1	F_2	mAP_{50}	$\text{mAP}_{50:95}$
VOC_v	LIG*	7,340	0.716	0.898	0.760	0.729	0.522	0.328
	TTN_D	19,265	0.682	**0.513**	**0.512**	**0.576**	**0.485**	**0.308**
	TTN	22,084	0.686	0.483	0.490	0.562	0.470	0.300
	AL [15]	47,139	**0.715**	0.425	0.442	0.528	0.438	0.281
COCO_v	LIG*	89,503	0.422	0.898	0.545	0.461	0.204	0.124
	TTN_D	446,132	0.413	0.392	0.338	**0.355**	0.184	0.112
	TTN	319,843	0.380	**0.410**	**0.340**	0.345	**0.185**	**0.113**
	AL	544,203	**0.421**	0.381	0.333	0.353	0.175	0.106
LVIS_v	LIG*	266	0.027	0.269	0.045	0.032	0.0004	0.0003
	TTN_D	27,025	**0.027**	**0.034**	**0.013**	**0.014**	0.0007	0.0006
	TTN	29,297	0.026	0.032	**0.013**	0.013	**0.0009**	**0.0007**
	AL	31,402	**0.027**	0.032	0.012	**0.014**	0.0007	0.0006
BDD_v	LIG*	453	0.077	0.576	0.123	0.091	0.036	0.016
	TTN_D	18,535	0.075	**0.074**	**0.035**	**0.042**	**0.013**	**0.006**
	TTN	32,899	0.076	0.035	0.022	0.031	0.006	0.003
	AL	34,888	**0.077**	0.036	0.022	0.032	0.007	0.003
Overall_v	LIG*	24,390	0.311	0.660	0.368	0.328	0.191	0.117
	TTN_D	127,739	0.299	**0.253**	**0.225**	**0.247**	**0.171**	**0.107**
	TTN	101,031	0.292	0.240	0.216	0.238	0.166	0.104
	AL	164,408	**0.310**	0.218	0.202	0.231	0.155	0.098

on these failing categories. Remarkably, **TTN_D approximately doubles precision, F_1, mAP_{50}, and $\text{mAP}_{50:95}$** on BDD_v with only a 3% reduction in recall after pruning 53% of the labels, as shown in Tab. 5. Across all datasets, zero-shot TTN and TTN_D improve Overall $\text{mAP}_{50:95}$ by 6% and 9%, respectively.

By selectively pruning disruptive candidates, both TTNs facilitate a particularly striking **Category Revival**, where the downstream model achieves non-trivial mAP on classes that previously exhibited zero recall. For example, at a $\tau = 0.1$ threshold on LVIS, the AL baseline exhibits 0 recall for the 259 vulnerable classes regardless of VLM. The zero-shot TTN recovers recall for 37 of these categories (27, 10 for YOLOE, YOLOW), while TTN_D recovers 28 (17, 11 for YOLOE, YOLOW). This disproportionate improvement suggests that many

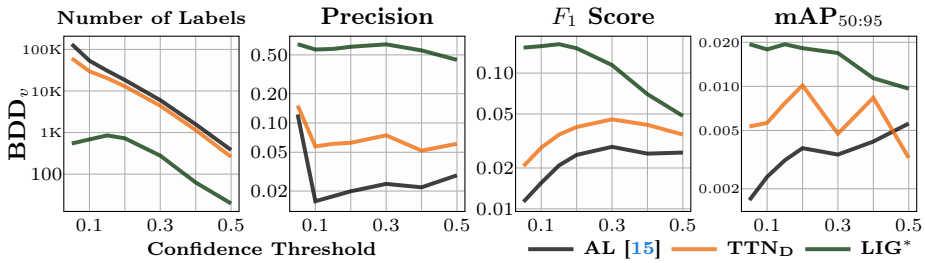

Fig. 7: The Post-Engine Interrogation Frontier (log scale). The pruning gap on BDD_v (left) shows the *sheer magnitude* of VLM hallucinations that the LIG* Oracle successfully discards, whereas even high-confidence filtering retains these errors. Precision, F_1 score, and $\text{mAP}_{50:95}$ performance across confidence thresholds τ (right) shows that TTN_D significantly narrows the performance gap in the high-noise, low-τ regime.

transfer-vulnerable classes do not suffer from a lack of data, but rather from an exceptionally high noise-to-signal ratio. This recovery of the most challenging classes shows that the TTN's semantic pruning is a critical component for achieving robust generalization in fully autonomous pseudo-labeling pipelines.

7 Conclusion and the Post-Engine Interrogation Frontier

In this paper, we introduced the Label Imitation Game (LIG), a Turing-inspired framework that formalizes pseudo-label pruning as a dataset-wide adversarial interrogation. By developing the Turing Test Network (TTN), we demonstrate that a model trained strictly on image classification can effectively judge and "detoxify" complex object detection pseudo-labels. Our experiments across four diverse datasets show that the TTN successfully suppresses systemic foundation model hallucinations, yielding zero-shot F_1-score gains of 28% for the most challenging categories—rising to 44% with task-specific fine-tuning (TTN_D). Crucially, both models facilitate Category Revival, enabling downstream detectors to recover from zero recall on dozens of transfer-vulnerable classes. Our results establish that global semantic-contextual logic is a robust alternative to local spatial-geometric verification. By releasing our LIG framework and pre-trained TTN weights, we provide the vision community with a lightweight, task-agnostic module that enhances pseudo-labeling rigor at scale.

Looking forward, a significant **Post-Engine Interrogation Frontier** remains for future work. As shown in Fig. 7, the optimal pruning regime includes low-confidence labels, where the most authentic signal is often discarded by traditional filtering. TTN_D substantially narrows this performance gap, but additional gains must be won from increasingly subtle and difficult-to-identify hallucinations. We postulate that integrating multi-model consensus and LIG-aware spatial redundancy reduction will close this gap. To broaden applicability, expanding the TTN's interrogative capacity to video [7] and 3D modalities [38] provides new frontiers for fully autonomous, noise-robust machine learning.

References

1. Bogdoll, D., Ananta, R.P., Giridharan, A., Moore, I., Stevens, G., Liu, H.X.: Mcity data engine: Iterative model improvement through open-vocabulary data selection. arXiv preprint 2504.21614 (2025) 3
2. Boonsirisumpun, N., Okafor, E., Surinta, O.: Vehicle image datasets for image classification. Data in Brief (2024) 8
3. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: European Conference on Computer Vision (ECCV) (2014) 8
4. Bras, R.L., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M., Sabharwal, A., Choi, Y.: Adversarial filters of dataset biases. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020) 3
5. Chen, C., Debattista, K., Han, J.: Pseudo-labelling should be aware of disguising channel activations. In: European Conference on Computer Vision (ECCV) (2024) 3
6. Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y.: Yolo-world: Real-time open-vocabulary object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3, 9
7. Dave, I.R., Rizve, M.N., Shah, M.: Finepseudo: Improving pseudo-labelling through temporal-alignablity for semi-supervised fine-grained action recognition. In: European Conference on Computer Vision (ECCV) (2024) 14
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 8
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (IJCV) (2010) 9
10. Freire, A., de S. Silva, L.H., de Andrade, J.V., Azevedo, G.O., Fernandes, B.J.: Beyond clean data: Exploring the effects of label noise on object detection performance. Knowledge-Based Systems (2024) 3
11. Fu, S., Yang, Q., Mo, Q., Yan, J., Wei, X., Meng, J., Xie, X., Zheng, W.S.: Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) (2025) 1
12. Gao, X., Zhang, Y., Galley, M., Dolan, B.: An adversarially-learned turing test for dialog generation models. arXiv preprint arXiv:2104.08231 (2021) 3
13. Geman, D., Geman, S., Hallonquist, N., Younes, L.: Visual turing test for computer vision systems. Proceedings of the National Academy of Sciences (PNAS) (2015) 3
14. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014) 3
15. Griffin, B.A., Gangwar, M., Sela, J., Corso, J.J.: Auto-labeling data for object detection. arXiv preprint arXiv:2506.02359 (2025) 1, 2, 3, 10, 11, 12, 13, 14
16. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. In: International Conference on Learning Representations (ICLR) (2022) 1
17. Gui, Z., Sun, S., Li, R., Yuan, J., An, Z., Roth, K., Prabhu, A., Torr, P.: kNN-CLIP: Retrieval enables training-free segmentation on continually expanding large vocabularies. Transactions on Machine Learning Research (TMLR) (2024)

18. Gunjal, A., Yin, J., Bas, E.: Detecting and preventing hallucinations in large vision language models. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024) 1, 3
19. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 9
20. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems (NeurIPS) (2018) 1, 3
21. He, Y., Chen, W., Liang, K., Tan, Y., Liang, Z., Guo, Y.: Pseudo-label correction and learning for semi-supervised object detection. arXiv preprint arXiv:2303.02998 (2023) 3
22. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019) 8
23. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
24. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: Proceedings of the 35th International Conference on Machine Learning (ICML) (2018) 1, 3
25. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics. https://github.com/ultralytics/ultralytics (2023)
26. Jocher, G., Qiu, J.: Ultralytics yolo11. https://github.com/ultralytics/ultralytics (2024) 11
27. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr - modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
28. Krizhevsky, A.: Learning multiple layers of features from tiny images (Technical Report, 2009) 8
29. Lee, D.H.: Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop : Challenges in Representation Learning (WREPL) (2013) 1
30. Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., Liu, Y.: Llms-as-judges: A comprehensive survey on llm-based evaluation methods. arXiv preprint arXiv:2412.05579 (2024) 3
31. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., Gao, J.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2
32. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (2023) 1, 3
33. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: The European Conference on Computer Vision (ECCV) (2014) 9
34. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: The European Conference on Computer Vision (ECCV) (2024) 3, 9

35. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 1
36. Moore, B.E., Corso, J.J.: Fiftyone. https://github.com/voxel51/fiftyone (2020) 2
37. Nagase, Y., Babazaki, Y., Shibata, T.: Annotation-free object detection by knowledge-extraction training from visual-language models. In: International Conference on Pattern Recognition (ICPR) (2025) 1, 2
38. Ošep, A., Meinhardt, T., Ferroni, F., Peri, N., Ramanan, D., Leal-Taixé, L.: Better call sal: Towards learning to segment anything in lidar. In: European Conference on Computer Vision (ECCV) (2024) 14
39. Park, D., Choi, S., Kim, D., Song, H., Lee, J.G.: Robust data pruning under label noise via maximizing re-labeling accuracy. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 3
40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 7
41. Popp, N., Zhang, D., Metzen, J.H., Hein, M., Schott, L.: Single-pass object-focused data selection. arXiv preprint arXiv:2412.10032 (2025) 1, 2
42. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 8
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML) (2021) 6
44. Salscheider, N.O.: Featurenms: Non-maximum suppression by learning feature embeddings. In: 2020 25th International Conference on Pattern Recognition (ICPR) (2021) 3
45. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
46. Sheng, M., Sun, Z., Chen, T., Pang, S., Wang, Y., Yao, Y.: Foster adaptivity and balance in learning with noisy labels. In: European Conference on Computer Vision (ECCV) (2024) 3
47. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 3
48. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950) 2, 4
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 5
50. Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., Ding, G.: Yoloe: Real-time seeing anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025) 9
51. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush,

A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics (2020)
52. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
53. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017) 8
54. Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z.: End-to-end semi-supervised object detection with soft teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
55. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics (2014)
56. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 9
57. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 3
58. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2020) 2
59. Zhou, Y., Cui, C., Yoon, J., Zhang, L., Deng, Z., Finn, C., Bansal, M., Yao, H.: Analyzing and mitigating object hallucination in large vision-language models. In: The Twelfth International Conference on Learning Representations (ICLR) (2024) 1, 3