

Article

AgenticInspect: Risk-Aware Segmentation Refinement with Hybrid Auditing for Electronic Interconnection Inspection in Smart Manufacturing

Yun Lin ¹, Yu-Shan Jiang ¹, Dai-Yan Ji ², Ruoxin Wang ^{2,*} and Jay Lee ²¹ Foxconn Interconnect Technology, New Taipei City 236038, Taiwan; yun.y.lin@fit-foxconn.com (Y.L.); connie.ys.jiang@fit-foxconn.com (Y.-S.J.)² Industrial AI Center, University of Maryland, College Park, MD 20742, USA; jidn@umd.edu (D.-Y.J.); leejay@umd.edu (J.L.)

* Correspondence: rwang135@umd.edu

Abstract

Industrial visual inspection requires not only accurate segmentation but also reliable and traceable decision making, since incorrect automatic refinements may lead to downstream manufacturing failures. Although segmentation foundation models achieve strong segmentation accuracy, they provide limited support for assessing whether refinement actions should be accepted, rejected, or escalated before deployment. To address this challenge, we propose AgenticInspect, a hybrid-audited agentic inspection framework that reformulates segmentation refinement as an action-level, risk-aware decision process. AgenticInspect treats an edge-deployable segmentation model as a replaceable skill. Based on the output of the segmentation skill, AgenticInspect introduces three task-specific decision modules, including uncertainty- and geometry-guided candidate generation, train-learned gating for high-risk refinement actions, and hybrid audit for risk-controlled mask selection. The hybrid audit layer integrates deterministic safety rules, train-learned safety gates, and selective large language model (LLM) review. The proposed framework provides action-level risk control and auditable decision making, with a traceable rationale for accepting, rejecting, or escalating candidate refinements. AgenticInspect was evaluated on a labeled connector-inspection dataset. The proposed refinement pipeline improved IoU from 0.9189303 to 0.9190180. The full hybrid policy reduced degraded refinements from 14 to 12 while maintaining nearly all performance gains (IoU = 0.9190131). Audit-component ablation attributes the observed reduction in IoU-degrading outputs to the train-learned risk veto, while LLM review serves as a selective, structured, and auditable assistive interface rather than an independently validated source of automatic improvement. These results position AgenticInspect as a risk-aware inspection framework that converts static segmentation outputs into auditable refinement decisions.

Keywords: Agentic AI; segmentation refinement; risk-aware decision making; hybrid audit; visual inspection; connector manufacturing

Academic Editor: George A. Tsihrintzis

Received: 20 August 2026

Revised: 8 October 2026

Accepted: 13 October 2026

Published: 16 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the rapid development of electric vehicles [1], aerospace electronics [2], data centers [3], and 5G/6G communication systems [4], interconnection structures are

becoming increasingly compact, high-density, and design-diverse. Electronic interconnection manufacturing plays a fundamental role in signal and power transmission in electronic systems. In the electronic interconnection manufacturing process, conductor alignment inspection is a critical vision task before welding, because the relative position between the exposed conductor and the target joining region directly affects contact area, joining stability, and downstream process quality. Especially, conductor alignment inspection remains challenging in real-world production environments. The target conductive structures are usually small, reflective, and sensitive to variations in wire posture, terminal geometry, surface condition, illumination, and imaging setup. Conductor misplacement may lead to insufficient overlap, asymmetric joining, repeated process adjustment, or rework. In addition, product upgrades and design changes can introduce substantial variations in conductor count, spatial layout, and visual appearance. Traditional manual inspection is inefficient and difficult to standardize, while rule-based machine vision methods are often sensitive to threshold settings, background interference, and product-specific geometric assumptions. These limitations motivate deep-learning-based visual inspection methods that can localize conductor regions and extract alignment-related geometric features from industrial images.

Segmentation methods have been regarded as potential technology because pixel-level masks can provide detailed features, including conductor area, centerline, spacing, boundary position, and overlap with the intended joining region. Recent segmentation foundation models [5] and lightweight edge-deployable variants [6] have further improved the feasibility of mask generation in industrial applications. In practice, however, raw segmentation outputs are often refined using adaptive thresholding, morphological operations, connected-component filtering, and task-specific geometric constraints before being used for downstream inspection decisions. Although these refinement operations can improve average segmentation quality, they may also degrade individual samples by removing valid conductor regions, expanding masks into background areas, or altering fine geometric boundaries. Such sample-level degradations are particularly risky when segmentation outputs are directly connected to automatic alignment judgment or process feedback.

Therefore, conductor alignment inspection requires more than a strong segmentation model. It requires a risk-aware refinement mechanism that can determine whether a candidate mask should be trusted, whether proposed refinement should be rejected, and whether an uncertain case should be escalated for human review. This motivates a shift from static mask generation to auditable segmentation decision making. Instead of treating post-processing as a fixed sequence of operations, the inspection system should maintain an explicit inspection state, evaluate candidate refinements, control high-risk actions, and provide traceable reasons for accepting or rejecting a refined mask. To address this problem, we propose *AgenticInspect*, a hybrid-audited agentic segmentation refinement framework for risk-aware smart inspection that wraps a frozen segmentation model as a replaceable skill and augments it with stateful planning, candidate refinement, hybrid auditing, fallback control, and interpretable inspection traces. Our contributions can be concluded three-fold as follows:

1. We introduce a compact refinement pipeline consisting of uncertainty-aware base refinement, geometry-guided candidate generation, and train-learned risky-action gating.
2. We design a selective hybrid audit layer that combines deterministic safety rules, train-learned risk veto, and local LLM-based structured review for accept/reject/human-review decisions.

3. We provide five-fold cross-validation and audit-component ablations showing that hybrid auditing reduces harmful degraded cases while preserving most of the mean-IoU gain introduced by action-gated risky refinement.

2. Related Work

2.1. Industrial Visual Inspection and Deployment Reliability

Industrial visual inspection requires more than high average predictive accuracy. In electronic interconnection manufacturing, localized segmentation errors may propagate into geometric measurements, alignment estimation, or downstream pass–fail decisions. Therefore, occasional severe failures can be more consequential than small changes in mean intersection over union (IoU). Existing studies on printed circuit board and electrical connector inspection have demonstrated the value of deep learning for automated defect localization and component analysis [7]. Industrial anomaly-detection benchmarks have similarly encouraged pixel-level evaluation under realistic manufacturing conditions [8].

Nevertheless, most existing inspection approaches optimize detection or segmentation performance while treating the generated output as the final system response. Limited attention has been given to determining whether an automatically generated refinement should be accepted, rejected, or replaced by a safer fallback. This distinction is important because a refinement operation may improve some masks while degrading others. Consequently, reliable deployment requires explicit control over refinement actions rather than unconditional post-processing. AgenticInspect addresses this deployment gap by formulating segmentation refinement as a risk-aware inspection decision problem in which candidate actions are evaluated before they affect the final inspection output.

2.2. Promptable Segmentation Foundation Models as Inspection Skills

The Segment Anything Model (SAM) introduced promptable segmentation through point, box, and mask prompts and demonstrated strong transferability across diverse visual domains [5]. Its general-purpose mask-generation capability makes it attractive as a reusable perception component in industrial inspection systems. However, the original SAM relies on a computationally expensive vision-transformer encoder, which limits its direct use in latency- or resource-constrained manufacturing environments.

Several efficient SAM variants have therefore been developed. MobileSAM replaces the original image encoder with a lightweight encoder trained through decoupled distillation [9]. FastSAM reformulates segment-anything inference using a convolutional instance-segmentation architecture [10]. EdgeSAM introduces a selective knowledge distillation method for deployment on edge devices [11], while EfficientSAM employs masked-image pretraining to obtain lightweight promptable segmentation models [12]. These developments substantially improve the accessibility of promptable segmentation for real-time and edge applications.

Despite their efficiency, these models primarily generate masks in response to prompts. They do not inherently determine whether a task-specific refinement is beneficial, whether a resulting geometry is plausible, or whether the refined output should be trusted for downstream manufacturing decisions. In AgenticInspect, the frozen segmentation foundation model is therefore treated as a replaceable inspection skill, rather than as the complete inspection system. The primary contribution lies in controlling and auditing how this skill is invoked and how its candidate outputs are selected.

2.3. Uncertainty-Aware Refinement and Selective Decision Making

Segmentation refinement commonly employs threshold adjustment, morphological operations, connected-component filtering, hole filling, and removal of small regions.

These procedures can correct local defects and enforce task-specific geometric priors, but their effectiveness depends on the input mask and object geometry. An operation that improves an incomplete mask may over-expand or distort an already accurate prediction. Consequently, fixed post-processing pipelines cannot guarantee monotonic improvement for every sample.

Uncertainty estimation provides one mechanism for identifying unstable predictions. Bayesian SegNet estimates pixel-level uncertainty through Monte Carlo dropout [13], while deep ensembles provide a scalable approach to predictive uncertainty and distribution-shift awareness [14]. However, high confidence does not necessarily imply correctness because modern neural networks can produce poorly calibrated probabilities [15]. Confidence-derived features should therefore be interpreted as decision evidence rather than as absolute reliability guarantees.

Selective prediction extends this principle by allowing a model to abstain from predictions for which the expected risk is high. SelectiveNet jointly learns prediction and selection functions to control the trade-off between coverage and predictive risk [16]. Selective segmentation further adapts this idea to dense prediction, where uncertain pixels or samples can be withheld, referred, or processed through alternative mechanisms [17]. These studies demonstrate that practical deployment objectives may differ from maximizing average segmentation accuracy over all inputs.

AgenticInspect is related to selective prediction but operates at the refinement-action level. Instead of deciding only whether to issue a segmentation prediction, it evaluates whether a candidate thresholding or geometry-based action should replace the existing mask. The system can preserve the current result, accept a candidate refinement, or invoke a safer fallback. This formulation connects uncertainty-aware segmentation with risk-controlled action selection and enables the effects of automatic refinement to be evaluated through beneficial, unchanged, and harmful outcomes.

2.4. Agentic and Hybrid-Audited Inspection Workflows

Agentic AI systems organize models and external tools into structured perception, reasoning, action, and feedback workflows. ReAct demonstrated that interleaving reasoning traces with tool actions can improve task execution and interpretability [18]. Reflexion introduced feedback-based reflection over previous outcomes [19], while Self-Refine showed how generated outputs can be iteratively evaluated and revised using explicit feedback [20]. These methods establish the broader concept of treating an intermediate model output as a state that can be examined and acted upon rather than as an immutable final answer.

Agent-based approaches have also begun to appear in manufacturing applications. Large-language-model agents have been used to invoke deterministic process-planning tools and execute multistep manufacturing workflows [21], while recent multi-agent manufacturing systems use modular agents to support adaptive shop-floor decisions [22]. However, unrestricted language-model reasoning is difficult to justify as the sole safety mechanism for precision inspection because its decisions may be nondeterministic and difficult to validate.

AgenticInspect therefore adopts a bounded and task-specific form of agency. The framework invokes specialized segmentation and refinement skills, represents intermediate masks and geometric measurements as explicit states, records action decisions, and audits candidate refinements before deployment. Its hybrid audit combines deterministic constraints, a learned risk veto, and an optional language-model-based interpretation layer. The language model provides contextual explanation and escalation support, whereas the primary deployment safeguards remain grounded in measurable mask properties and validated decision policies. This design distinguishes AgenticInspect from both

unconditional segmentation pipelines and unconstrained general-purpose agents, and the resulting functional differences are summarized in Table 1.

Table 1. Functional comparison between conventional segmentation methods and AgenticInspect.

Method	Segmentation	Refinement	Risk Control	Explainable	Human Review
SAM	✓	-	-	-	-
MobileSAM	✓	-	-	-	-
Morphology	-	✓	-	-	-
CRF	-	✓	-	-	-
Agent Framework	✓	✓	Partial	Partial	Optional
AgenticInspect (Ours)	✓	✓	✓	✓	✓

3. Methodology

3.1. Problem Formulation

Given an input image I , a frozen segmentation skill S produces a probability map:

$$P = S(I). \quad (1)$$

Applying a predefined fixed threshold τ_0 yields the raw segmentation mask:

$$M_0 = \mathbb{1}[P \geq \tau_0]. \quad (2)$$

An uncertainty-aware threshold policy π_T selects an image-specific threshold τ_i and produces the adaptive-threshold mask:

$$M_A = \mathbb{1}[P \geq \tau_i], \quad \tau_i = \pi_T(P). \quad (3)$$

Based on M_A , the candidate-generation stage produces conservative geometry-guided candidates and higher-risk morphology-based candidates. A geometry selector produces a conservative geometry-safe mask M_G , while the train-learned action gate may select a candidate refinement mask M_C associated with an action a_C :

$$M_G = \pi_G(\mathcal{C}_G(M_A)), \quad (4)$$

$$(M_C, a_C) = \pi_A(\mathcal{C}_R(M_A)), \quad (5)$$

where $\mathcal{C}_G$ and $\mathcal{C}_R$ denote the conservative and higher-risk candidate sets, respectively.

The hybrid audit policy evaluates the selected candidate using structural constraints, a train-learned risk veto, and optional LLM-assisted review:

$$d = \pi_H(M_A, M_G, M_C, a_C, \mathbf{x}), \quad (6)$$

where $\mathbf{x}$ denotes the probability-, mask-, and geometry-derived decision features, and

$$d \in \{\text{AUTO_ACCEPT}, \text{AUTO_REJECT}, \text{HUMAN_REVIEW}\}. \quad (7)$$

Under the main hybrid policy, the final automatic mask is defined as follows:

$$M^* = \begin{cases} M_G, & \text{if no higher-risk action is selected,} \\ M_C, & \text{if } d = \text{AUTO_ACCEPT,} \\ M_A, & \text{if } d = \text{AUTO_REJECT,} \\ M_A, & \text{if } d = \text{HUMAN_REVIEW,} \end{cases} \quad (8)$$

where HUMAN_REVIEW additionally triggers escalation for manual inspection. Alternative deployment policies may instead use M_G as a fallback mask.

A conventional segmentation system is commonly evaluated by maximizing average segmentation quality:

$$\max_{\pi} \frac{1}{N} \sum_{i=1}^N \text{IoU}(M_i^*, G_i). \quad (9)$$

However, mean IoU alone may conceal sample-level degradation caused by automatic refinement. We therefore define the change relative to the fixed-threshold raw mask as follows:

$$\Delta_i^{(0)} = \text{IoU}(M_i^*, G_i) - \text{IoU}(M_{0,i}, G_i). \quad (10)$$

To isolate the effect of post-threshold candidate refinement and auditing, we additionally define the following:

$$\Delta_i^{(A)} = \text{IoU}(M_i^*, G_i) - \text{IoU}(M_{A,i}, G_i). \quad (11)$$

Given a numerical tolerance ϵ , each sample is categorized as Equation (12):

$$\begin{aligned} \text{Better}_i &= \mathbb{1}[\Delta_i > \epsilon] \\ \text{Same}_i &= \mathbb{1}[|\Delta_i| \leq \epsilon] \\ \text{Worse}_i &= \mathbb{1}[\Delta_i < -\epsilon] \end{aligned} \quad (12)$$

Degradation rate is calculated as Equation (13):

$$R_{\text{deg}} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}[\Delta_i < -\epsilon] \quad (13)$$

Worst observed drop is defined as Equation (14):

$$D_{\text{worst}} = \min_i \Delta_i \quad (14)$$

AgenticInspect is therefore evaluated using complementary objectives: preserving or improving mean segmentation quality while reducing the frequency and severity of harmful automatic refinements. The framework does not claim a formal safety guarantee; instead, it provides empirical action-level risk control through selective acceptance, rejection, and fallback.

3.2. AgenticInspect Framework

Figure 1 illustrates the overall architecture of AgenticInspect. The framework consists of six stages: input image, frozen segmentation-skill inference, uncertainty-aware threshold selection, candidate generation, hybrid auditing, and final output generation. Intermediate features and decisions are maintained in the inspection state and recorded in the decision trace to support subsequent refinement and audit decisions.

Given an input connector image, the frozen segmentation skill first produces a probability map (P). In our implementation, the segmentation skill is instantiated using EdgeSAM exported to ONNX. The uncertainty-aware threshold gate then selects an image-specific threshold according to probability-derived uncertainty and context features, producing the adaptive-threshold mask (M_A).

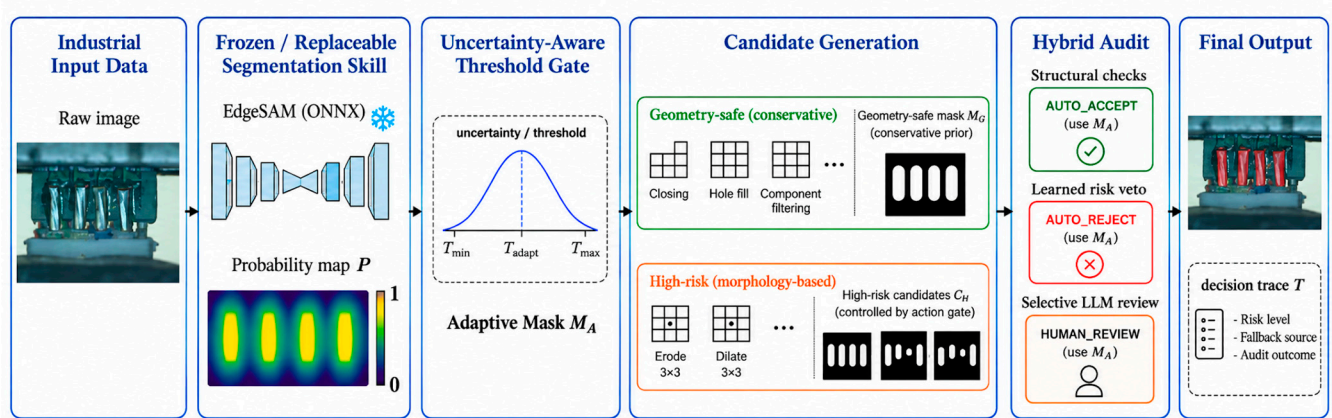


Figure 1. Overall framework of AgenticInspect. The framework consists of five stages: segmentation, uncertainty-aware mask generation, candidate generation, hybrid auditing, and final decision making. Ambiguous predictions are refined through candidate masks and evaluated using both rule-based verification and LLM-assisted review before producing the final output.

The candidate-generation stage constructs two types of refinement proposals from the adaptive-threshold mask (M_A). Conservative geometry-guided operations, such as closing, hole filling, and component filtering, are used to obtain a geometry-safe mask (M_G). In parallel, higher-risk morphology-based candidates, including erosion and dilation, are generated and controlled by the train-learned action gate before they can be considered for final selection.

If no high-risk action is selected, the geometry-safe mask (M_G) is retained. Otherwise, the action-gated high-risk candidate is forwarded to the hybrid audit module for risk-controlled selection. The audit combines deterministic structural checks, a train-learned risk veto, and selective LLM review. Depending on the audit outcome, the candidate may be automatically accepted, automatically rejected, or escalated for human review. Under the main hybrid policy, rejected or escalated high-risk candidates fall back to the adaptive-threshold mask (M_A).

The framework finally outputs the selected segmentation mask together with a decision trace (T). The trace records the audit outcome, risk level, and fallback source, providing an interpretable record of how the final inspection result was obtained.

Figure 2 provides a detailed view of the hybrid audit and fallback policy. An action-gated high-risk candidate is first evaluated by deterministic structural checks and a train-learned risk veto. Candidates that fail either stage are automatically rejected and revert to the adaptive-threshold mask (M_A). Candidates that pass both stages are either automatically accepted or, when additional review is required, forwarded to the selective LLM auditor. The LLM can return AUTO_ACCEPT, AUTO_REJECT, or HUMAN_REVIEW. Accepted candidates are used as the final mask, whereas rejected candidates fall back to M_A ; HUMAN_REVIEW also preserves M_A as the automatic output while escalating the case for manual inspection. The resulting decision and fallback information are recorded in the decision trace.

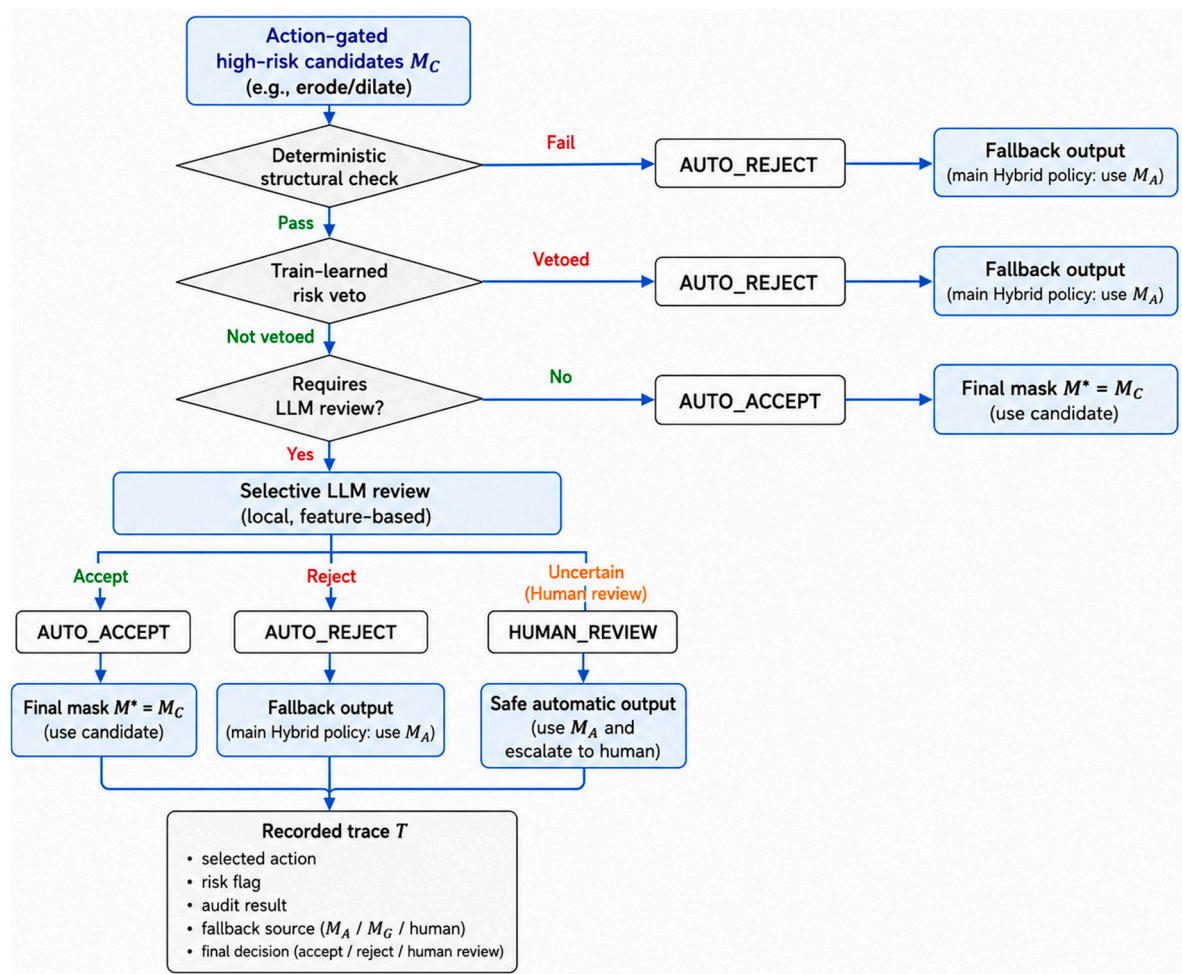


Figure 2. Hybrid audit and fallback policy.

3.3. Uncertainty-Aware and Geometry-Guided Candidate Generation

The candidate-generation stage produces a set of refinement proposals based on both probabilistic uncertainty information and task-specific geometric priors. As shown in Figure 1, the process consists of four steps: segmentation-skill inference, uncertainty-aware threshold selection, candidate refinement generation, and geometry-based candidate scoring. The resulting candidates are subsequently evaluated by the train-learned action gate and the hybrid audit module.

3.3.1. Segmentation Skill

In our implementation, we instantiate the segmentation skill using EdgeSAM exported to ONNX. The deployed model adopts an EfficientViT-B0 encoder with a binary segmentation head, and the checkpoint achieving the highest validation IoU is selected. The model uses an input resolution of 1024 and is kept frozen throughout all experiments. Input images are converted to RGB, scaled to [0,1], and resized to the configured input resolution using bilinear interpolation, without ImageNet mean/std normalization. The segmentation skill then performs ONNX inference, converts the output logits to probabilities using a sigmoid function, resizes the probability map to the original image resolution, and applies the subsequent thresholding strategy to obtain the initial mask. Importantly, thresholding is performed after probability resizing rather than on the low-resolution output before resizing.

Although we use EdgeSAM in this study, AgenticInspect is not tied to a specific segmentation model. Other SAM-like models, task-specific segmentors, or probability-producing segmentation networks can be used as alternative segmentation skills.

3.3.2. Uncertainty-Aware Threshold Gate

The threshold gate selects a binarization threshold using features extracted from the probability map. Candidate thresholds are evaluated on the training split, and simple rules are learned to choose when to deviate from the default threshold. The threshold features include probability statistics and threshold-sensitivity descriptors, such as the following:

- Probability mean and quantiles.
- Foreground area at candidate thresholds.
- Area drop between thresholds.
- Uncertain probability bands.

The gate is fitted exclusively using the training samples of each cross-validation fold and is applied to the corresponding held-out samples without using their ground-truth labels.

3.3.3. Geometry Scoring

After obtaining an initial mask, AgenticInspect generates a set of candidate refinements. To rank candidate refinements, AgenticInspect computes a geometry score using multiple structural indicators derived from the mask geometry. These indicators include component count, area consistency, spacing consistency, size consistency, center distance, and noise penalty. Component count measures agreement with the expected number of pin components. Area consistency evaluates the similarity of component areas. Spacing consistency measures the regularity of adjacent pin spacing. Size consistency evaluates variation in component size. Center distance measures deviations of component centroids from the expected layout. Noise penalty discourages isolated artifacts and spurious components. Candidate refinements with higher geometry scores are preferred during geometry-guided refinement. These include conservative operations:

- Small-component removal.
- Hole filling.
- Morphological opening and closing.
- Keeping the largest expected components.
- Combinations of closing, cleaning, and component filtering.

The candidate set also includes higher-risk actions such as erosion and dilation. These actions can improve masks in some cases but may also damage boundary quality or component structure. Therefore, they are controlled by the train-learned action gate and the audit module.

3.4. Train-Learned Gating for High-Risk Refinement Actions

The action gate determines when high-risk operations such as `erode_3 × 3` and `dilate_3x3` are allowed. For each training sample, candidate action features are computed relative to the initial mask, and the action's IoU change is measured using the training ground truth. The gate then learns simple univariate rules that identify cases where a risky action is likely to help.

For the main EfficientViT-B0 1024 model on the development training split, the action gate learns two rules:

- `erode_3 × 3`: `delta.mask_ratio` ≥ -0.0012749466 .
- `dilate_3 × 3`: `delta.compactness` ≤ 0.0005842897 .

These rules are reported as interpretable examples from the development analysis and can be audited directly. In the five-fold cross-validation experiments, the action-gating rules were re-estimated independently using only the training samples of each fold. Only candidates satisfying the learned gating criteria were allowed to proceed to the hybrid audit stage.

3.5. Hybrid Audit for Risk-Controlled Mask Selection

The audit skill reviews the selected refinement and produces:

- AUTO_ACCEPT.
- AUTO_REJECT.
- HUMAN_REVIEW.

The hybrid policy applies three audit layers. First, deterministic safety checks reject structural violations such as unexpected component changes, added border contact, or increased fragments and holes. Second, a train-learned risk veto detects interpretable high-risk action patterns that produce degradation on training candidates. Third, only unresolved erosion or dilation proposals are sent to a local LLM auditor. The LLM auditor uses a fixed instruction template with dynamically inserted structured inputs, including the initial and refined mask features, selected action, candidate quality scores, refinement attempts, gating information, and the rule-based fallback audit result. The prompt specifies the expected four-component pin topology and conservative acceptance criteria and requires a structured JSON output of AUTO_ACCEPT, AUTO_REJECT, or HUMAN_REVIEW. The detailed format is listed in Appendix A.

The strict automatic policy accepts only AUTO_ACCEPT. Under the main hybrid policy, both AUTO_REJECT and HUMAN_REVIEW return the adaptive-threshold mask M_A as the automatic output. HUMAN_REVIEW additionally triggers manual escalation. Alternative geometry-safe fallback policies using M_G are evaluated in Section 5.2. This design separates refinement exploration from deployment-safe behavior and limits LLM involvement to ambiguous risky actions.

AgenticInspect outputs the final mask, overlay artifacts, extracted features, selected actions, audit decisions, risk levels, and an agent trace. This trace makes the inspection process auditable: users can see which skill was invoked, which action was selected, and why a refinement was accepted or rejected.

4. Experiments Setup

4.1. Dataset

We evaluate AgenticInspect on a pin segmentation inspection task containing 238 labelled images. The dataset consists of two naming conventions: 100 images indexed as 1–100 and 138 images indexed as 001_0321–138_0321. These identifiers correspond to distinct inspection samples and should not be interpreted as belonging to the same instance (e.g., 001 and 001_0321 represent different samples). Each image corresponds to a distinct inspection sample.

For formal evaluation, we use five-fold cross-validation, producing held-out folds containing 48, 48, 48, 47, and 47 images. The dataset contains two subsets with different naming conventions (1–100 and 001_0321–138_0321). Fold assignment was performed to ensure that samples from both subsets were represented in each fold, thereby maintaining a similar data composition across the evaluation splits. In each fold, threshold, action, and risk-veto gates are learned using only the corresponding training folds, and performance is reported on the unseen fold.

The previously explored 190/48 split was used during methodological development for diagnostic analysis, including candidate-operation design, feature exploration, and

module-level evaluation. Because these samples are also included in the subsequent five-fold cross-validation dataset, results from this split are reported only as exploratory evidence and should not be interpreted as independent validation. The cross-validation results are therefore treated as out-of-fold internal validation of the finalized pipeline rather than as evaluation on a fully untouched external test set.

4.2. Evaluation Metrics

To evaluate both segmentation quality and refinement reliability, we report the mean intersection over union (IoU), IoU gain relative to the comparison baseline, Better/Same/Worse counts, and the worst observed IoU drop.

Mean IoU measures the overall segmentation quality across the evaluation set. To assess sample-level refinement behavior, the IoU change for each sample is categorized as Better, Same, or Worse according to a numerical tolerance ϵ . Two comparison baselines are used. First, end-to-end performance is evaluated relative to the fixed-threshold raw segmentation output M0, which captures the cumulative effect of the complete AgenticInspect pipeline. Second, post-threshold refinement is evaluated relative to the adaptive-threshold mask MA, allowing the effects of geometry refinement, action gating, and hybrid auditing to be isolated from improvements introduced by adaptive threshold selection. Better/Same/Worse are therefore defined using per-sample IoU changes and quantify pixel-overlap improvement or degradation rather than task-specific geometric reliability. Qualitative geometry analysis is used separately to examine cases in which IoU improvement may not correspond to more suitable inspection geometry.

The number of worse cases are used to quantify the frequency of harmful automatic refinements, while the worst observed IoU drop measures the largest sample-level degradation. These metrics complement mean IoU by distinguishing average segmentation improvement from refinement risk at the individual-sample level.

4.3. Implementation Details

We use ONNX Runtime for segmentation inference. We evaluate EfficientViT-B0, MobileNetV3-small, ResNet18, and ResNet34 variants at 512 and 1024 input resolutions on the development split. EfficientViT-B0 1024 achieves the strongest raw IoU in that comparison and is therefore selected as the backbone configuration for the EdgeSAM segmentation skill used in the cross-validation experiments of the full policy. The predefined baseline threshold is 0.45. Candidate morphology uses small kernels such as 3×3 , 5×5 , and 7×7 . In cross-validation, threshold, action, and veto rules are retrained from scratch within each training fold only. The threshold gate searches for single-feature rules over probability-derived features to select thresholds from 0.20 to 0.80. Rules require at least five training samples, a mean IoU gain above 0.0005, and at most two degraded cases; up to four rules are retained. The action gate learns separate geometry-based rules for 3×3 erosion and dilation, requiring at least eight samples, a mean gain above 0.0002, and at most three degraded cases. Eligible rules are ranked by training IoU gain with penalties for degradation. The risk veto searches for up to two-condition rules among action-gate-permitted candidates, retaining rules that prevent net IoU loss without rejecting improving training cases. At inference, unmatched threshold rules retain the default threshold of 0.45, while vetoed refinements revert to the adaptive input mask. All rules are fitted using only the training folds. For selective hybrid auditing, local Ollama LLMs are queried deterministically with temperature zero and a fixed seed.

5. Results

5.1. Main Results

Table 2 reports the five-fold cross-validation results over 238 out-of-fold predictions. The complete AgenticInspect pipeline consists of adaptive thresholding, geometry-guided refinement, a train-learned action gate, and hybrid auditing. The LLM-based configurations use deterministic judgments generated by qwen2.5:7b-instruct. Adaptive thresholding contributes most of the accuracy gain, while geometry refinement improves four additional out-of-fold cases without introducing additional degraded cases. The action gate obtains the highest mean IoU but increases degraded cases from eight to fourteen. LLM-only auditing accepts all selected high-risk proposals in this evaluation and therefore does not change the action-gate result. In contrast, hybrid rejects five risk-flagged proposals and reduces degraded cases from fourteen to twelve, at a small IoU cost of 0.0000049 relative to the action-gate policy. This establishes hybrid primarily as a risk-mitigation layer. Here, ϵ are set as 10^{-12} for Better/Same/Worse classification. The reduction in degraded cases persists at tolerances of 10^{-5} and 10^{-4} , indicating that it is not solely attributable to numerical precision. However, both methods yield eight degraded cases at 10^{-3} , showing that the observed advantage in degradation counts depends on the tolerance used to define degradation.

To separate the effects of refinement from those of adaptive threshold selection, Table 3 compares all post-threshold refinement policies using the same adaptive-threshold output as the baseline. The controlled comparison shows that conservative geometry refinement introduces no additional degradation after adaptive thresholding. The hybrid policy retains most of the action-gate gain while reducing the number of harmful post-threshold refinements from six to four relative to both the action-gate-only and LLM-only policies. Figure 3 further illustrates the sample-dependent effects of morphology-based refinement and the limitations of IoU as the sole evaluation criterion. In the first two examples, 3×3 dilation substantially reduces IoU, whereas 3×3 erosion improves the segmentation outputs. In the third example, both operations increase IoU, with dilation producing the larger numerical improvement from 0.9041 to 0.9326. However, the dilated mask introduces geometrically undesirable boundary expansion that may bias downstream measurements of conductor width, spacing, or alignment. This example demonstrates that a higher IoU does not necessarily correspond to a more reliable mask for industrial inspection. Consequently, evaluating refinement quality solely according to IoU may favor masks that are numerically improved but operationally unsuitable for downstream inspection. This discrepancy motivates the joint consideration of pixel-level overlap and task-specific geometric consistency through geometry-aware action selection and risk-aware auditing.

Table 2. Comparison performance of AgenticInspect and baseline models.

Adaptive Threshold	Geometry	Action Gate	Audit	IoU	Δ IoU vs. Raw	Better/Same/Worse	Worst-Case Δ IoU
–	–	–	–	0.9189303	0	0/238/0	0
✓	–	–	–	0.9189860	+0.0000556	12/218/8	−0.0041466
✓	✓	–	–	0.9189953	+0.0000650	16/214/8	−0.0041466
✓	✓	✓	–	0.9190180	+0.0000877	22/202/14	−0.0058220
✓	✓	✓	LLM-only	0.9190180	+0.0000877	22/202/14	−0.0058220
✓	✓	✓	Hybrid	0.9190131	+0.0000828	23/203/12	−0.0058220

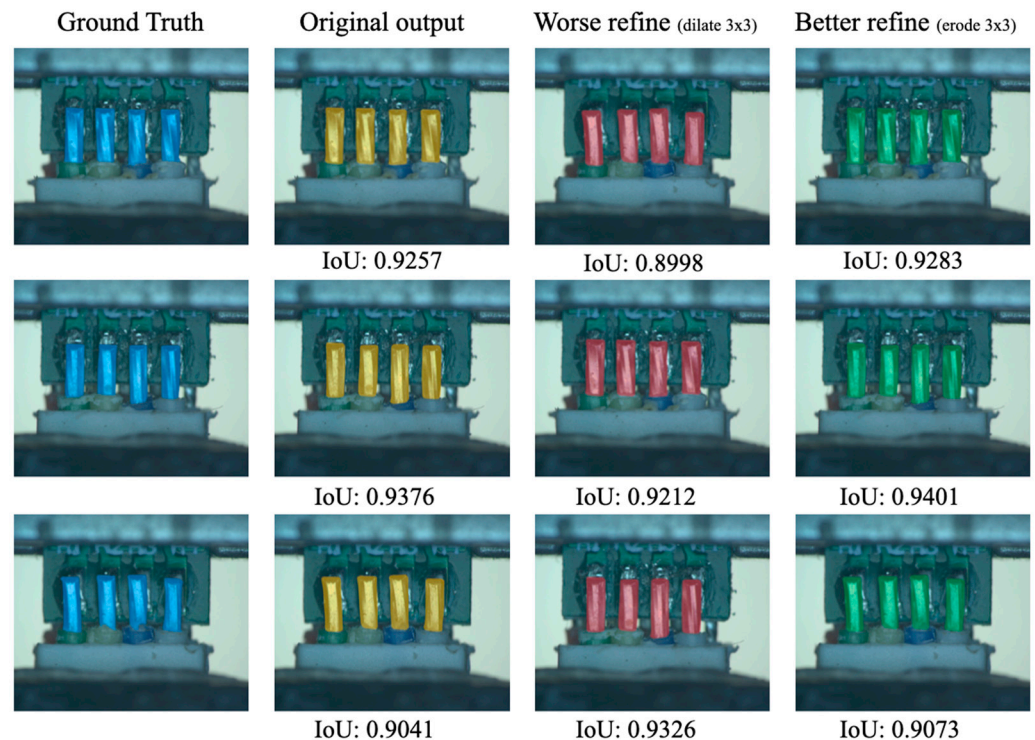


Figure 3. Qualitative comparison of morphology-based segmentation refinement outcomes. From left to right: ground-truth masks, original segmentation outputs, 3×3 dilation candidates, and 3×3 erosion candidates. The examples demonstrate that the same morphological operation may improve or degrade segmentation quality across different samples, motivating risk-aware action selection and auditing.

Table 3. Effect of refinement policies after adaptive threshold selection.

Geometry	Action Gate	Audit	Δ IoU vs. Adaptive	Better/Same/Worse
✓	–	–	+0.0000093	5/233/0
✓	✓	–	+0.0000320	15/217/6
✓	✓	LLM-only	+0.0000320	15/217/6
✓	✓	Hybrid	+0.0000271	12/222/4

5.2. Audit Component and Safe-Fallback Ablation

Table 4 summarizes the audit-component ablation results under cross-validation. To determine which audit component contributes to the reduction in harmful automatic refinements, different decision-layer variants are evaluated on the same set of action-gated refinement proposals. All reported values are computed from out-of-fold samples.

Table 4. Audit component ablation study results.

Audit Policy After Action Gate	IoU	Gain vs. Raw	Better/Same/Worse vs.	
			Raw	Adaptive
No Audit	0.9190180	+0.0000877	22/202/14	15/217/6
Hard Rule Only	0.9190180	+0.0000877	22/202/14	15/217/6
LLM Only	0.9190180	+0.0000877	22/202/14	15/217/6
Risk Veto Only	0.9190131	+0.0000828	23/203/12	12/222/4
Risk Veto + LLM	0.9190131	+0.0000828	23/203/12	12/222/4
Full Hybrid: Hard Rule + Risk Veto + LLM	0.9190131	+0.0000828	23/203/12	12/222/4

The ablation shows that the reduction in harmful automatic outputs is supplied by the train-learned risk veto. Hard-rule and LLM-only audits do not alter any selected action under cross-validation. The full hybrid policy therefore preserves a structured LLM audit interface, but the measured automatic risk reduction in this experiment should be attributed specifically to veto rules.

In addition to the audit-component analysis, we investigate the effect of different fallback strategies when high-risk refinement proposals are rejected. Table 5 compares alternative fallback policies using the same cross-validation setting.

Table 5. Comparison of fallback policies for rejected high-risk refinements.

Automatic Policy	IoU	Gain vs. Raw	Better/Same/Worse vs. Raw	Better/Same/Worse vs. Adaptive
Adaptive Threshold + Geometry Safe Base	0.9189953	+0.0000650	16/214/8	5/233/0
Current Hybrid: Rejected Action to Adaptive	0.9190131	+0.0000828	23/203/12	12/222/4
Hybrid-Safe: Rejected Action to Geometry	0.9190131	+0.0000828	23/203/12	12/222/4
Strict-Safe: All High-Risk Actions to Geometry	0.9189953	+0.0000650	16/214/8	5/233/0

Fallback to geometry does not change hybrid in the current evaluation because the geometry-selected output equals the adaptive mask on all five vetoed cases. A strict deployment policy that routes every erosion/dilation proposal to the geometry-safe output eliminates all post-adaptive degradation but also gives up the additional action-gate gain.

5.3. Development-Split Module Combination Ablation

Using the previously explored 190/48 development split, we further test valid combinations of the proposed modules for diagnostic analysis. The results are listed in Table 6. These combinations follow the framework dependency structure: an action gate operates on geometry-generated candidates, and hybrid auditing requires a proposed refinement to review. Consequently, combinations such as action gate alone or hybrid audit alone are not operationally meaningful.

Table 6. Development-split ablation of AgenticInspect module combinations.

Adaptive Threshold	Geometry	Action Gate	Hybrid Audit	Val IoU	Gain vs. Raw	Better/Same/Worse
-	-	-	-	0.9167390	0.0000000	0/48/0
-	✓	-	-	0.9167396	+0.0000006	1/47/0
-	✓	-	✓	0.9167396	+0.0000006	1/47/0
-	✓	✓	-	0.9167851	+0.0000461	4/42/2
-	✓	✓	✓	0.9168671	+0.0001281	4/43/1
✓	-	-	-	0.9169703	+0.0002313	8/36/4
✓	✓	-	-	0.9169709	+0.0002319	9/35/4
✓	✓	-	✓	0.9169709	+0.0002319	9/35/4
✓	✓	✓	-	0.9169898	+0.0002508	8/33/7
✓	✓	✓	✓	0.9170044	+0.0002654	9/33/6

The development-split combination study provides three observations. First, adaptive thresholding is the dominant contributor to mean IoU. Second, hybrid auditing has no measurable effect when it receives only conservative geometry proposals: all such

proposals are automatically accepted. Third, hybrid auditing becomes useful once the action gate introduces higher-risk proposals. These observations motivate the formal cross-validation risk comparison in Sections 5.1–5.3 rather than serving as final generalization evidence.

To distinguish independently executable modules from dependent decision modules, we report standalone or minimum-operational configurations, as shown in Table 7. Adaptive thresholding and geometry refinement can each transform a raw mask independently. In contrast, the action gate requires generated candidates, and hybrid audit requires a candidate decision to audit.

Table 7. Contribution of individual AgenticInspect modules under minimum-operational configurations.

Module of Interest	Standalone Executable	Evaluated Configuration	Val IoU	Gain vs. Raw	Better/Same/Worse
Adaptive Threshold	✓	Adaptive Threshold only	0.9169703	+0.0002313	8/36/4
Geometry-Guided Refinement	✓	Geometry only	0.9167396	+0.0000006	1/47/0
Train-Learned Action Gate	-	Geometry + Action Gate	0.9167851	+0.0000461	4/42/2
Hybrid Audit	-	Geometry + Action Gate + Hybrid Audit	0.9168671	+0.0001281	4/43/1

Among genuinely standalone processing modules, adaptive thresholding supplies the largest gain, while geometry refinement supplies a small but degradation-free improvement. The action gate and hybrid audit should not be interpreted as standalone segmentation operators: their reported minimum-operational configurations include the candidate-generation stages required for them to act.

5.4. Segmentation-Skill and Input-Resolution Ablation

We conduct the segmentation-skill comparison on the 48-image development split. To compare framework portability fairly across all skills, AgenticInspect in Table 8 applies the shared geometry-based candidate refinement module to each raw mask; formal cross-validation full-policy results for EfficientViT-B0 1024 are reported in Section 5.1. The 1024 skills outperform their 512 counterparts, confirming the importance of resolution for fine pin structures. EfficientViT-B0 1024 obtains the highest raw IoU in this development comparison and supports its selection as the skill used in cross-validation. The portable geometry-only refinement is highly conservative.

Table 8. Segmentation-skill and input-resolution comparison with conservative geometry refinement on the development split.

Model	Input Size	Raw IoU	Geometry-Refined IoU	Gain	Better/Same/Worse
EfficientViT-B0	512	0.8773247	0.8773250	+0.0000002	1/47/0
MobileNetV3-small	512	0.8410514	0.8410513	−0.0000000	0/47/1
ResNet18	512	0.7720336	0.7720336	0.0000000	0/48/0
ResNet34	512	0.8051811	0.8051811	−0.0000000	0/47/1
EfficientViT-B0	1024	0.9167390	0.9167396	+0.0000006	1/47/0
MobileNetV3-small	1024	0.9150902	0.9150902	0.0000000	0/48/0
ResNet18	1024	0.8967968	0.8968175	+0.0000206	1/47/0
ResNet34	1024	0.8988614	0.8988614	0.0000000	0/48/0

5.5. Threshold Hyperparameter Ablation

We ran a fixed-threshold diagnostic evaluation for the EfficientViT-B0 1024 skill on the 48-image development split. The results are listed in Table 9. The development-set sweep peaks at 0.50, only 0.0000078 IoU above the predefined 0.45 baseline. Lower thresholds increase recall but reduce precision, while higher thresholds increase precision but reduce recall. We retained 0.45 as the predefined fixed raw reference used by the train-learned pipeline.

Table 9. Threshold sensitivity analysis of the EfficientViT-B0 1024 segmentation skill.

Threshold	Val IoU	Precision	Recall
0.20	0.9145175	0.9420393	0.9691008
0.25	0.9153942	0.9449578	0.9670077
0.30	0.9159763	0.9474442	0.9650622
0.35	0.9163852	0.9496780	0.9632058
0.40	0.9166429	0.9517360	0.9613799
0.45	0.9167390	0.9536487	0.9595399
0.50	0.9167468	0.9554832	0.9576972
0.55	0.9166741	0.9572879	0.9558097
0.60	0.9164714	0.9590878	0.9538001
0.65	0.9161141	0.9608800	0.9516467

5.6. Development-Split Audit Case Study

For interpretability, we retain a development-split case study comparing rule-only, LLM-only, and hybrid decisions from the same adaptive-threshold input mask. The LLM configurations use deterministic qwen2.5:7b-instruct generation. The results are listed in Table 10. Hybrid improves over LLM-only by 0.0000562 IoU and reduces degraded refinement cases from three to two on this development split. Specifically, LLM-only accepts a harmful dilate_3x3 action on 125_0321, whereas the train-learned veto rule initial.pin_spacing_cv ≤ 0.0274411 rejects it, recovering 0.0026975 IoU for that sample. Here, pin_spacing_cv measures the relative variability of adjacent pin spacings, and it can be calculated as Equation (17):

$$\text{pin_spacing_cv} = \frac{\sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (d_i - \bar{d})^2}}{\bar{d}} \quad (17)$$

where n denotes the number of pins, d_i denotes the spacing between adjacent pins, and $\bar{d}$ denotes the mean spacing. This case illustrates the mechanism behind the broader cross-validation finding that hybrid reduces harmful risky-action acceptance.

Table 10. Development-split comparison of audit policies.

Audit Policy	Val IoU	Gain vs. Adaptive Input	Better/Same/Worse	Worst Drop
Rule-only	0.9169703	0.0000000	0/48/0	0.0000000
Action gate without audit	0.9169898	+0.0000195	3/42/3	−0.0034441
LLM-only	0.9169482	−0.0000221	2/43/3	−0.0034441
Hybrid rule + veto + LLM	0.9170044	+0.0000341	2/44/2	−0.0034441

5.7. Remaining Failure Analysis

On the development split, despite its improvement over LLM-only, hybrid still accepts two harmful refinements relative to its adaptive-threshold input. The first occurred

on sample 102_0321, where a dilate_3x3 operation decreased IoU by 0.0034441. The second occurred on sample 126_0321, where an erode_3x3 operation decreased IoU by 0.0009737. Furthermore, four end-to-end worse cases on this split are already introduced before the audit stage by adaptive threshold selection. This pattern is consistent with cross-validation, where adaptive thresholding produces eight worse out-of-fold cases and the full hybrid pipeline does not eliminate them.

The remaining failure samples can be broadly categorized into two groups. First, some degradations originate from the adaptive-threshold selection stage before any morphology refinement is applied. In these cases, the initial mask quality is already compromised, and the subsequent audit mechanism cannot fully recover the error because no high-risk morphology action is responsible for the degradation.

Second, a small number of harmful refinements remain after action selection and auditing. Visual inspection suggests that these samples typically exhibit ambiguous boundary confidence, irregular conductor geometries, or local spacing patterns that resemble the characteristics of previously beneficial refinements. Consequently, they may not be effectively identified by the current risk-veto criteria. These observations suggest that additional geometric descriptors, boundary-stability measures, or topology-consistency indicators may further improve discrimination between beneficial and harmful refinements in challenging cases.

6. Discussion

The experiments show that the main value of AgenticInspect is not a large increase in mean segmentation accuracy. The frozen segmentation model is already strong, leaving limited room for post-processing gains. Instead, AgenticInspect provides a structured mechanism for exposing and managing refinement risk.

Cross-validation clarifies the role of each layer. Adaptive thresholding and geometry produce conservative improvement. Action gating increases mean IoU but also introduces additional degraded cases. Risk veto only and full hybrid obtain identical automatic outputs: both reduce the action-gated degradation count from fourteen to twelve while retaining most of its mean gain. Across four valid-output local LLM auditors, LLM-only review accepts all high-risk proposals and contributes no measurable automatic decision change beyond veto rules. The supported claim is therefore narrower and stronger: AgenticInspect provides selective, auditable LLM-assisted review around a quantitatively validated veto-based risk-control core.

The framework is also model-agnostic. EdgeSAM is used as the segmentation skill in our experiments, but the same agentic design can wrap other SAM-like or task-specific segmentors. The backbone ablation shows that the quality of the segmentation skill remains important, and that 1024 input resolution is critical for fine pin structures.

There are several limitations. First, although all train-learned rules are fitted exclusively within the training portion of each cross-validation fold, the overall framework and several methodological choices were developed through earlier exploration of the same 238-image dataset. Therefore, the five-fold cross-validation results should be interpreted as out-of-fold internal validation rather than fully independent external validation. The dataset also remains modest and does not include a separately collected untouched test set. Second, the geometry prior is tailored to four-pin structures and would need to be modified for other inspection objects. Third, the current refinement candidates are morphology-based and cannot correct large semantic segmentation errors. Finally, hybrid reduces but does not eliminate harmful automatic changes, and adaptive threshold selection itself contributes degraded cases.

The reported performance trends should be interpreted within the scope of the connector-inspection dataset studied. Other electronic-interconnection inspection tasks may

exhibit different pin counts, object scales, surface reflectance characteristics, imaging conditions, and defect modes. Consequently, the quantitative improvements reported in this work may not directly transfer to other inspection scenarios without task-specific adaptation and validation.

7. Conclusions

We presented AgenticInspect, an agentic inspection framework for risk-aware segmentation refinement. AgenticInspect treats a segmentation model as a replaceable skill and augments it with stateful planning, adaptive thresholding, geometry-guided candidate refinement, action gating, train-learned risk veto, selective LLM auditing, and interpretable trace output. Under five-fold cross-validation, the action-gated pipeline obtains the highest mean IoU, while veto-controlled hybrid reduces degraded out-of-fold cases from fourteen to twelve with a small mean-IoU tradeoff. Component and model ablations show that the observed reduction in segmentation degradation arises from the learned veto rather than from LLM-only judgment. LLM auditing remains useful as a selective, traceable interface for assistive review, while stricter policies remain necessary for fully automatic deployment.

Author Contributions: Conceptualization: Y.L. and R.W.; data curation: Y.L.; methodology: Y.L. and R.W.; formal analysis and investigation: Y.L. and D.-Y.J.; project administration and funding acquisition: J.L.; software, validation, and visualization: Y.L.; writing—original draft: Y.L.; writing—review and editing: J.L. and R.W.; supervision: R.W.; resources: Y.-S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by funding from the Industrial AI Center of University of Maryland College Park (2-957942).

Data Availability Statement: The datasets generated and/or analyzed during this study are not publicly available due to confidentiality and proprietary restrictions related to real-world manufacturing data from Foxconn Interconnect Technology. However, the data may be made available from the corresponding author upon reasonable request and subject to approval by the data provider.

Acknowledgments: The authors acknowledge Foxconn Interconnect Technology (FIT) for facilitating access to the manufacturing environment and data used in this study. During the preparation of this study, the authors used Claude sonnet 4.5 for grammar and logic revision. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: Author Yun Lin and Yu-Shan Jiang are employees of Foxconn Interconnect Technology. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

```

1.  Role:
2.  You are an industrial segmentation refinement auditor.
3.
4.  Task:
5.  Decide whether to accept the proposed mask refinement.
6.
7.  Constraints:
8.  - Expect four independent pin-like components.
9.  - Reject changes that worsen fragmentation, holes, border
10.   contact, or cause large area changes.
11. - Preserve the initial mask when improvement is uncertain.
12. - Request human review when automatic acceptance is unsafe.
13.
14. Input:
15. {
16.   "initial_features": {...},
17.   "refined_features": {...},
18.   "selected_action": "...",
19.   "quality_score": ...,
20.   "candidate_attempts": [...],
21.   "rule_based_audit": {...}
22. }
23.
24. Output:
25. {
26.   "accept_refinement": true | false,
27.   "risk_level": "LOW | MEDIUM | HIGH",
28.   "review_decision": "AUTO_ACCEPT | AUTO_REJECT | HUMAN_REVIEW",
29.   "reason": "...",
30.   "suggestion": "..."
   }

```

Figure A1. The template of the LLM auditor.

References

1. Paret, P.; Finegan, D.; Narumanchi, S. Artificial intelligence for power electronics in electric vehicles: Challenges and opportunities. *J. Electron. Packag.* **2023**, *145*, 034501.
2. Baruah, R.K.; Yoo, H.; Lee, E.K. Interconnection technologies for flexible electronics: Materials, fabrications, and applications. *Micromachines* **2023**, *14*, 1131.
3. Xie, C.; Zhang, B. Scaling optical interconnects for hyperscale data center networks. *Proc. IEEE* **2022**, *110*, 1699–1713.
4. Khalid, R.; Raza, M.N. Analyzing the impact of optical wireless communication technologies on 5G/6G and IoT solutions: Prospects, developments, and challenges. *J. Eng. Res. Sci.* **2024**, *3*, 23–36.

5. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 4015–4026.
6. Chen, L.; Wang, X.; Yao, F.; Song, M.; Zhang, J.; Wang, S. An edge-guided SAM for effective complex object segmentation. *Expert Syst. Appl.* **2025**, *271*, 126560.
7. Alharthi, S.S.; Fallatah, A.M.; Al-Saidi, H.M. Design and Characterization of Electrochemical Sensor for the Determination of Mercury(II) Ion in Real Samples Based upon a New Schiff Base Derivative as an Ionophore. *Sensors* **2021**, *21*, 3020.
8. Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9592–9600.
9. Zhang, C.; Han, D.; Qiao, Y.; Kim, J.U.; Bae, S.H.; Lee, S.; Hong, C.S. Faster segment anything: Towards lightweight SAM for mobile applications. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 2404–2413.
10. Zhao, X.; Zhang, C.; Chaudhuri, A.; Dey, D.; Xiang, T.; Torr, P. Fast segment anything. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
11. Wang, R. EdgeSAM. GitHub Repository. 2026. Available online: <https://github.com/meredith92/EdgeSAM> (accessed on 8 October 2026).
12. Xiong, Y.; Varadarajan, B.; Wu, L.; Xiang, X.; Xiao, F.; Zhu, C.; Dai, X.; Wang, D.; Sun, F.; Iandola, F.; et al. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16111–16121.
13. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 4–7 September 2017.
14. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; p. 30.
15. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
16. Geifman, Y.; El-Yaniv, R. SelectiveNet: A deep neural network with an integrated reject option. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 2151–2159.
17. Ding, Y.; Liu, J.; Xu, X.; Huang, M.; Zhuang, J.; Xiong, J.; Shi, Y. Uncertainty-Aware Training of Neural Networks for Selective Medical Image Segmentation. *Proc. Mach. Learn. Res.* **2020**, *121*, 156–173.
18. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing reasoning and acting in language models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
19. Shinn, N.; Cassano, F.; Labash, B.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 8634–8652.
20. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46534–46594.
21. Holland, M.; Chaudhari, K. Large language model based agent for process planning of fiber composite structures. *Manuf. Lett.* **2024**, *40*, 100–103.
22. Zhao, Z.; Tang, D.; Liu, C.; Wang, L.; Zhang, Z.; Zhu, H.; Chen, K.; Nie, Q.; Ji, Y. A large language model-based multi-agent manufacturing system for intelligent shopfloors. *Adv. Eng. Inform.* **2026**, *69*, 103888.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.