# Agent-Ops: A Multi-Agent Orchestration Framework for End-to-End SOP Automation in E-Commerce Operations

**Apoorva Singh**[*]
Amazon, India
apoorvaz@amazon.com

**Sanjay Agrawal**[*]
Amazon, India
sanjagr@amazon.com

**Sayanta Adhikari**[*]
Amazon, India
sayantaa@amazon.com

**Vinayak Puranik**
Amazon, India
puranikv@amazon.com

**Shivam Tiwari**
Amazon, India
tishivak@amazon.com

**Dheeraj Assudani**
Amazon, India
dheeraa@amazon.com

## Abstract

While Large Language Models excel at reasoning and language understanding, they struggle with multi-step operational workflows requiring precise procedural adherence, which is fundamental for industrial automation. Existing SOP-guided agents assume well-defined procedures and structured APIs, failing to address enterprise realities like incomplete SOPs, dynamic web interfaces, and unpredictable document formats. We present Agent-Ops, an end-to-end multi-agent framework automating Standard Operating Procedures in e-commerce. Agent-Ops contributes: (1) **SOP Groomer**, a human-AI framework transforming ambiguous documentation into automation-ready specifications, (2) **WebAgent**, achieving 91.3% task completion and 86.5% execution consistency through demonstration-based learning, and (3) a **Document Verification Agent** performing multi-lingual validation across tax invoices, certificates, and supply chain documents with 94.2% accuracy. Deployed across seven SOP categories in three geographic regions, Agent-Ops achieves 85-97% end-to-end accuracy while reducing case resolution time by 83% reduction. Use across 100 Account Managers validates that LLM-based agents achieve enterprise-grade reliability when augmented with robust web automation, comprehensive document understanding, and systematic SOP refinement.

## 1 Introduction

Large Language Models (LLMs) have demonstrated impressive genral reasoning, language understanding, and problem solving abilities. However, they fall short in executing complex, multi-step workflows requiring strict adherence to Standard Operating Procedures (SOPs), which is critical for real-world industrial automation (Anonymous, 2025). This gap is particularly evident in knowledge operations work, where recent advances in agentic AI (Wang et al., 2024) have opened new frontiers.However, translating SOPs into reliable automated systems remains a significant enterprise challenge.

E-commerce marketplaces present a compelling case study. Account Managers (AMs) serve as the single point of contact for sellers, handling business advisory, operations (deals, pricing, inventory), and issue resolution. A typical AM manages multiple seller accounts. Among all responsibilities, product suppression management, where listings are temporarily removed due to policy violations, quality issues, or intellectual property concerns, represents one of the most time intensive tasks, consuming roughly 30 minutes per case.

The complexity in product suppression resolution illustrates the broader automation challenge. AMs must navigate over multiple internal systems including seller databases, account health tables, and government verification portals. Document validation requires authenticating tax invoices against eight criteria - date verification, quantity reconciliation, product name matching through fuzzy logic, and seller information cross-validation, across multiple languages (English, Hindi, Arabic, Portuguese, Turkish) and formats. For supply chain cases, complexity multiplies as AMs verify brand linkage, validate invoice sequences, and confirm distributor authenticity through comprehensive research. Each case requires interpreting nuanced policies varying by marketplace, product category, and suppression type.

Traditional automation approaches, including rule-based systems and browser automation tools like Selenium (He et al., 2024; Zheng et al., 2024; Deng et al., 2023), fail to address three barriers: (1) *Dynamic web interfaces* that change frequently, breaking hard-coded selectors, (2) *Multi-modal document processing* across diverse formats, languages, and quality levels, and (3) *Ambiguous pro-*

[*] These authors contributed equally to this research.

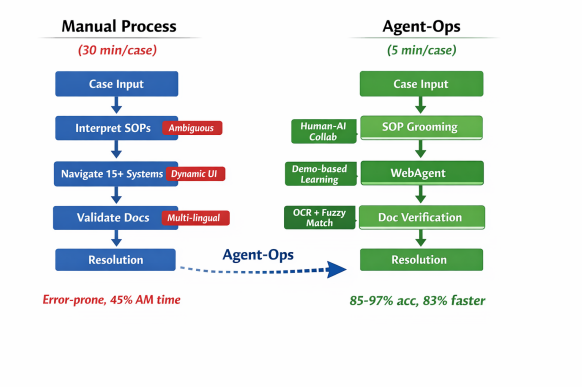


Figure 1: Agent-Ops workflow: Case Input → SOP Groomer → WebAgent → Document Verification, achieving 85-97% accuracy with 83% time reduction.

*cedural specifications* where SOPs assume domain expertise and omit critical details. While recent work has explored SOP guided agents using decision graphs (Ye et al., 2025) and multi-agent orchestration (Shrimal et al., 2024), these assume well defined procedures and structured APIs, failing to address enterprise realities.

Our work presents Agent-Ops, an end-to-end multi-agent framework, that bridges the gap between human authored SOPs and automated execution in e-commerce operations. This work makes three key contributions:

- **End-to-End Multi-Agent Framework**: Agent-Ops orchestrates specialized agents to automate complex operational workflows, addressing dynamic web interfaces, multi-modal document processing, and ambiguous procedural specifications.

- **Novel Technical Components**: We introduce: (1) **SOP Groomer**, which uses human-AI collaborative framework to transform ambiguous documentation into automation-ready specifications (+13.2% accuracy), (2) **WebAgent**, employing demonstration-based learning (91.3% task completion, 86.5% consistency), and (3) **Document Verification Agent**, performing multi-lingual document validation (94.2% accuracy).

- **Large-Scale Production Deployment**: We evaluate Agent-Ops on seven SOP categories across three regions, achieving 85-97% accuracy while reducing average case resolution time by 83%. Used across 100 Account Managers demonstrates practical viability on real-world cases with complex decision trees and cross-system data retrieval.

Our work validates that LLM-based agents can achieve enterprise-grade reliability when augmented with robust web automation, comprehensive document understanding, and systematic SOP refinement, offering a blueprint for automating knowledge operations across enterprise domains.

## 2 Related Work

Our work intersects four active research areas: LLM-based autonomous agents, SOP refinement, web automation, and document understanding.

### 2.1 LLM-Based Agents for Task Automation

Recent LLM advances enable autonomous agents for complex tasks through reasoning, planning, and tool use (Yao et al., 2023; Schick et al., 2023; Wei et al., 2022). MARCO (Shrimal et al., 2024) achieves 94.48% accuracy via multi-agent orchestration with validation guardrails, whileSOP-Agent (Ye et al., 2025) uses decision graphs achieving 86.6% HumanEval and 89.5% MBPP performance. Agent-S (Kulkarni, 2025) proposes workflows for customer care SOP automation, and SOP-Bench (Anonymous, 2025) establishes benchmarks revealing gaps in LLM capabilities for industrial SOPs. Multi-agent frameworks like AutoGen (Wu et al., 2023), MetaGPT (Hong et al., 2023), and CAMEL (Li et al., 2023) demonstrate collaborative architectures for task decomposition. These approaches assume well-defined procedures with structured APIs. Our work addresses enterprise realities: dynamic web interfaces, multi-modal documents, and ambiguous SOPs.

### 2.2 SOP Refinement and Workflow Generation

Human-authored SOPs contain implicit assumptions insufficient for automation. While program-synthesis (Chen et al., 2021; Li et al., 2022) and human-in-the-loop refinement (Settles, 2009; Wu et al., 2022) have been explored, systematic SOP refinement frameworks remainunderexplored. Our SOP Groomer provides structured refinement through step decomposition, RAG-based clarification (Lewis et al., 2020), multi-modal information extraction, and auto-merging, which improves automation accuracy by 13.2%.

### 2.3 Web Automation and Browser Agents

Autonomous web navigation faces challenges from dynamic interfaces (Shi et al., 2017). WebArena (Zhou et al., 2024) shows GPT-4 achieves below 15% success on complex tasks, while self-improvement approaches (Gur et al., 2024) demonstrate 31% gains. Browser agents like WebVoyager (He et al., 2024), SeeAct (Zheng et al., 2024), and Mind2Web (Deng et al., 2023) combine visual understanding with action prediction. WebGPT (Nakano et al., 2022) and RCI (Kim et al., 2023) explore browser-assisted question answering and computer task completion. Traditional tools (Selenium, Puppeteer, Playwright (Microsoft, 2020)) use hard-coded selectors that break with interface changes. Our WebAgent uses demonstration-based learning with three-stage element identification achieving 91.3% task completion and 86.5% consistency—crucial for production deployment.

### 2.4 Document Understanding and Verification

Document intelligence has evolved from OCR to multi-modal systems. LayoutLM (Xu et al., 2020) and successors (Xu et al., 2021; Huang et al., 2022; Kim et al., 2022) integrate textual and layout information, while DocFormer (Appalaraju et al., 2021), UDOP (Tang et al., 2023), and Pix2Struct (Lee et al., 2023) advance unified architectures. Vision-language models (GPT-4V (OpenAI, 2023), Gemini) enable zero-shot processing. Table understanding has progressed through models like TableFormer (Nassar et al., 2022), however enterprise verification requires multi-lingual support, cross-validation, fuzzy matching, and compliance checking. Our Document Verification Agent achieves 96.3% character accuracy (OCR), 93.7% field accuracy for tables, and 94.2% overall validation across jurisdictions.

## 3 Problem Statement

We formalize the SOP automation problem in enterprise e-commerce environments, characterizing the key challenges that motivate our multi-agent approach.

### 3.1 Task Definition

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ represent a Standard Operating Procedure comprising $n$ sequential steps. Each step $s_i$ involves: (1) *web navigation* actions $w_i$ requiring browser based interactions, (2) *document verification* tasks $d_i$ validating seller submitted materials across formats and languages, or (3) *decision points* $p_i$ determining case outcomes. Given input case $c$ with seller information, suppression details, and documents $D = \{d_1, d_2, \ldots, d_m\}$, the goal is to execute $\mathcal{S}$ to produce resolution decision $r \in \{\text{approve}, \text{reject}, \text{escalate}\}$ with supporting justification $j$.

### 3.2 Operational Complexity

Account Manager operations present four interconnected challenges - procedural ambiguity, system fragmentation, document heterogeneity, and decision complexity. These challenges are detailed in Appendix A.

### 3.3 Automation Requirements

Effective SOP automation must satisfy four requirements:

**Robustness**: Maintain consistent execution despite frequent interface changes, diverse document formats, and varying data quality. Traditional rule based automation with hard coded selectors fails within weeks due to interface updates.

**Accuracy**: Meet enterprise-grade thresholds (>85%) to minimize manual review overhead and maintain seller trust.

**Interpretability**: Provide clear justification with evidence sources, validation results, and reasoning steps for audit requirements and human oversight.

**Scalability**: Handle production workloads across multiple regions, SOP categories, and seller segments while maintaining low latency (<5 minutes per case) and supporting over 100 concurrent users.

Traditional rule based automation and general purpose LLM agents fail to address the dynamic, multimodal, and ambiguous nature of these operations, motivating our specialized multi-agent approach with systematic SOP refinement.

## 4 Tools and Capabilities Required

Effective SOP automation in e-commerce requires orchestrating specialized agents to handle complex operational workflows, as shown in Figure 2. Agent-Ops integrates autonomous agents and supporting capabilities for comprehensive end-to-end automation. We organize these into three categories: web automation, document intelligence, and compliance verification.

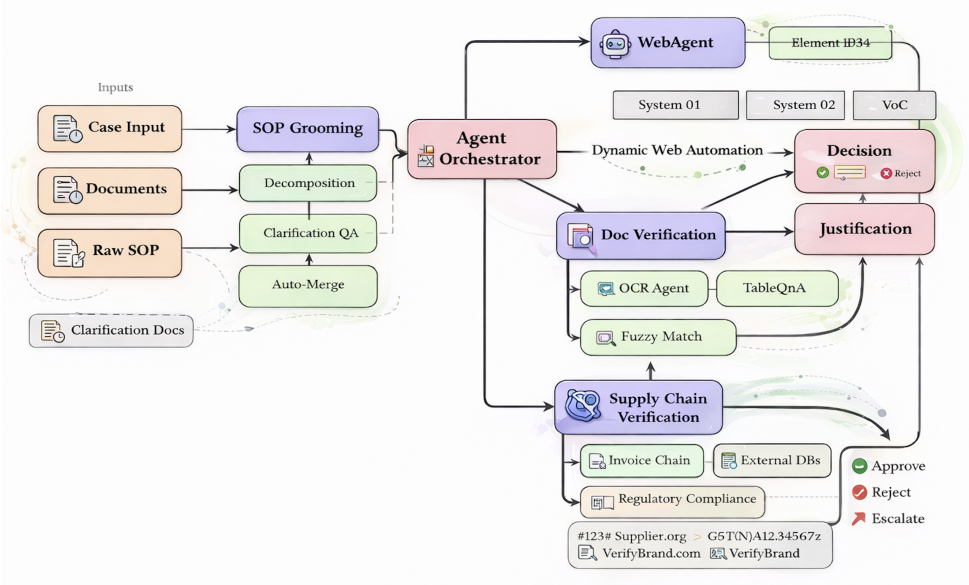


Figure 2: End-to-end SOP automation: Raw SOPs transformed via SOP Grooming, then executed by Agent Orchestrator coordinating web automation, document verification, and supply chain agents.

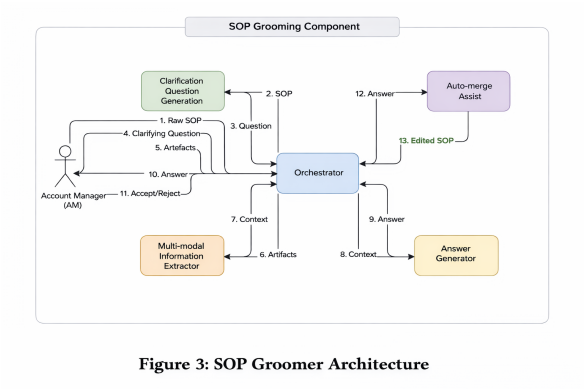


**Figure 3: SOP Groomer Architecture**

Figure 3: SOP Groomer: Raw SOPs decomposed → RAG based clarification questions → multimodal answer extraction → auto merged automation ready specifications.

### 4.1 Web Automation through WebAgent

WebAgent addresses navigation across 15+ internal systems with frequently changing interfaces. Unlike traditional browser automation tools (Selenium, Puppeteer) with brittle hardcoded selectors, WebAgent employs demonstration-based learning to generate robust automation templates. It implements a three stage Critical Element Identification: (1) Presence Verification using XPath and semantic matching, (2) Key-Value Signature Assignment identifying stable element characteristics, and (3) Configuration Persistence caching successful strategies. This maintains 86.5% execution consistency while adapting to interface changes.

### 4.2 Document Intelligence and Verification

Document validation requires multimodal processing across diverse formats and languages. Our framework integrates four capabilities: **OCR and Text Extraction** achieving 96.3% character accuracy across multiple languages, **Table Understanding** for SQL-like queries on structured data, **Document Classification** distinguishing tax invoices, LOA, certificates, and supply chain documentation, and **Fuzzy Matching** achieving 89.4% accuracy on seller name and product description matching.

### 4.3 Supply Chain and Compliance Verification

The Supply Chain Verification Agent validates multilevel invoice chains, integrates with external databases (government tax portals, commercial databases, brand authorization systems), and performs regulatory compliance checking across marketplace specific regulations varying by region, product category, and seller type.

## 5 SOP Groomer

SOPs authored by domain experts contain implicit assumptions and incomplete instructions insufficient for automated execution. Analysis reveals systematic gaps: references to systems without specification, validation instructions without criteria details, and decision points assuming marketplace specific policy knowledge. Our SOP Groomer framework addresses this through struc-

tured human-AI collaboration that transforms incomplete documentation into automation ready specifications.

### 5.1 Framework Architecture

SOP Groomer comprises four integrated components (Figure 3):

**SOP Decomposition** identifies discrete procedural steps while preserving hierarchical structure, classifying each by type (navigation, verification, decision) with metadata about required inputs and expected outputs.

**Clarification Question Generation** employs RAG (Lewis et al., 2020) to produce contextually relevant queries about underspecified instructions, targeting specific gaps like system fields and validation criteria.

**Multimodal Information Extraction** supports text documents (PDFs, wikis, training materials), browser replay sequences for navigation instructions, and screenshots for interface elements and workflow patterns, minimizing expert burden through automated answer generation.

**Auto Merge Integration** automatically integrates answers at precise ambiguity locations, preserving document structure while expanding underspecified instructions. A diff-based review interface allows expert validation before finalization.

All components are integrated in LangGraph (LangChain, 2024) orchestration for iterative refinement. See Appendix B for technical implementation details.

### 5.2 Grooming Effectiveness

We evaluated SOP Groomer on a set of representative SOPs spanning varying levels of procedural complexity. The grooming process automatically resolved the majority of underspecified or ambiguous steps through multi-modal information extraction, while a small fraction required targeted expert input for nuanced policy interpretation. SOPs refined using the grooming framework achieved a 13.2% relative improvement in automation accuracy compared to executing agents directly on raw procedural documentation, demonstrating the importance of systematic ambiguity resolution. In addition, the grooming framework substantially reduced the manual effort required for SOP preparation compared to fully manual rewriting approaches.

## 6 Experiments and Results

We evaluate Agent-Ops across individual component performance, end-to-end accuracy on representative SOP workflows, and overall system effectiveness under realistic operational conditions.

### 6.1 Experimental Setup

**Datasets.** We evaluate Agent-Ops on a collection of SOP categories across three anonymized regions ($R_A$, $R_B$, $R_C$), spanning varying levels of procedural complexity and compliance requirements, including product quality, authenticity verification, and intellectual property enforcement. The evaluation set consists of diverse case instances reflecting realistic variations in documentation quality, policy constraints, and decision complexity.

**Reproducibility and Hyperparameters:** We use a proprietary large language model as the underlying LLM across all agents. For details regarding reproducibility, hyperparameter configurations, and computational cost, please refer to Appendix E.

**Baselines.** For web automation: GPT-4 (zero-shot and few-shot) and rule-based Selenium. For document verification: standalone OCR pipelines (Tesseract, AWS Textract) and GPT-4V without specialized validation logic.

**Metrics.** We report: (1) Task Completion Rate: percentage of successful SOP executions; (2) Decision Accuracy: agreement with expert decisions; (3) Execution Consistency: percentage of repeated executions producing identical results; (4) Time Savings: reduction versus manual execution.

### 6.2 Web Automation Performance

Table 1 presents WebAgent's performance across internal seller sites, account health databases, and customer reviews and chats.

Table 1: Web Automation Task Completion Accuracy

| Method | Acc. (%) | Consist. (%) |
|---|---|---|
| GPT-4 (zero-shot) | 65.48 | 52.3 |
| GPT-4 (few-shot) | 71.82 | 61.5 |
| Rule-based Selenium | <u>78.34</u> | **91.2** |
| **WebAgent** | **91.3** | <u>86.5</u> |

WebAgent achieves 91.3% task completion, a 16.5% improvement over Selenium (78.34%). While Selenium achieves higher consistency (91.2%), it fails on 21.66% of cases due to interface changes. WebAgent's 86.5% consistency balances

reliability with adaptability—crucial for production deployment.

### 6.3 Document Verification Performance

The Document Verification Agent achieves 94.2% accuracy on invoice validation, significantly outperforming standalone OCR (76.3%) and GPT-4V (82.7%) lacking domain-specific validation. Key capabilities: Multi-lingual OCR (96.3% character accuracy), Field Extraction (93.7% accuracy), Fuzzy Matching (89.4% on seller/product matching), and Supply Chain Validation (87.2% on multi-level invoice chains).

### 6.4 End-to-End SOP Automation Results

Table 2 summarizes end-to-end automation performance across anonymized regions and representative SOP categories.

Table 2: End-to-End SOP Automation Accuracy Across Categories

| Region | SOP Category | Accuracy |
|---|---|---|
| $R_A$ | Product Quality | >90% |
| $R_B$ | Product Quality | 85–92% |
| $R_A$ | Intellectual Property | 97% |
| $R_C$ | Product Quality | 90% |

Across evaluated settings, Agent-Ops achieves end-to-end automation accuracy in the range of 85–97%. SOPs with more structured validation criteria, such as intellectual property enforcement, exhibit higher accuracy, while categories involving nuanced interpretation of documents and policies show comparatively lower performance. Performance variability across regions is primarily driven by differences in document heterogeneity and linguistic complexity.

### 6.5 Ablation Studies

**Critical Element Identification.** Removing three-stage element identification reduces WebAgent consistency from 86.5% to 58.3%, demonstrating importance of robust element localization.

**SOP Groomer.** Groomed SOPs achieve 13.2% higher accuracy than raw documents (88.1% vs. 74.9%). Analysis reveals ungroomed SOPs cause failures in navigation (42%), validation criteria (31%), and decision rules (27%).

**Multi-Agent Architecture.** Replacing specialized agents with single general-purpose agent reduces accuracy by 9.5% (88.1% to 78.6%), confirming value of task-specific design. General-purpose agent particularly struggles with document verification (15.2% drop) and web navigation consistency (12.8% drop).

## 7 Conclusion

We presented Agent-Ops, an end-to-end multi agent framework automating Standard Operating Procedures in e-commerce. Agent-Ops combines: **SOP Groomer** transforming ambiguous procedures into automation ready specifications (+13.2% accuracy), **WebAgent** for demonstration-based web automation (91.3% task completion), and **Document Verification Agent** for multilingual validation (94.2% accuracy).

Deployed across seven SOP categories in three regions, Agent-Ops achieves 85-97% accuracy while reducing case resolution by 83% (30 to 5 minutes). Key lessons: execution consistency builds trust, specialized agents outperform general approaches, and systematic SOP refinement is essential. Future work includes batch processing, expanded coverage, and continuous learning.

## 8 Limitations

While Agent-Ops demonstrates strong performance in SOP automation, our evaluation focuses on e-commerce marketplace operations. The framework supports adaptation to other domains but generalization to different operational contexts would require domain specific training data and validation rules. Also, our evaluation covers seven SOP categories in three geographical regions over a 3-month deployment period. Long-term performance degradation from policy or interface changes, or model staleness requires ongoing monitoring, hence continuous learning is needed as future work.

## References


Anonymous. 2025. Sop-bench: Evaluating llm agents on complex industrial sops. *Amazon Machine Learning Conference*. AMLC 2025.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. Self-improving web agents. *arXiv preprint arXiv:2404.07887*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. *European Conference on Computer Vision*, pages 498–517.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*.

Mandar Kulkarni. 2025. Agent-s: Llm agentic workflow to automate standard operating procedures. *arXiv preprint arXiv:2503.15520*.

LangChain. 2024. Langgraph: Building stateful, multi-actor applications with llms. *LangChain Documentation*.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. *International Conference on Machine Learning*, pages 18893–18912.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, and 1 others. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Microsoft. 2020. Playwright: Fast and reliable end-to-end testing for modern web apps. *GitHub Repository*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Christoph Auer. 2022. Tableformer: Table structure understanding with transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4614–4623.

OpenAI. 2023. Gpt-4v(ision) system card. *OpenAI Technical Report*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*.

Burr Settles. 2009. Active learning literature survey. *Computer Sciences Technical Report 1648, University of Wisconsin-Madison*.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. *International Conference on Machine Learning*, pages 3135–3144.

Anubhav Shrimal, Stanley Kanagaraj, Kriti Biswas, Swarnalatha Raghuraman, Anish Nediyanchath, Yi Zhang, and Promod Yenigalla. 2024. Marco: Multi-agent real-time chat orchestration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1381–1392.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264.

Shuai Wang, Zhe Li, Jing Zhao, and 1 others. 2024. A survey of multi-ai agent collaboration: Theories, frameworks, and applications. *ACM Computing Surveys*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. Human-in-the-loop machine learning: A survey. *arXiv preprint arXiv:2108.00941*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, and 1 others. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 2579–2591.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*.

Anbang Ye, Qianran Ma, Jia Chen, Muqi Li, Tong Li, Fujiao Liu, Siqi Mai, Meichen Lu, Haitao Bao, and Yang You. 2025. Sop-agent: Empower general purpose ai agent with domain-specific sops. *arXiv preprint arXiv:2501.09316*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Seeact: Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

## A Operational Complexity Details

Account Manager operations in e-commerce marketplaces present several interconnected challenges that make automation particularly difficult:

**Procedural Ambiguity.** Existing SOPs assume domain expertise and often omit critical details about tool usage, policy interpretation, and exception handling. SOPs may reference "check the system" without specifying which of 15+ systems to use, or state "validate the invoice" without detailing the eight validation criteria. This implicit knowledge, clear to experienced AMs, creates a significant barrier to automation.

**System Fragmentation.** Resolution of a single case requires navigation across 15+ internal systems including seller databases, account health databases, customer reviews and chats, and external government verification portals. Each system maintains distinct authentication mechanisms, interface patterns, data formats, and update frequencies. AMs must synthesize information across these fragmented systems to build a complete picture of each case, a process that currently takes 30 minutes per case on average.

**Document Heterogeneity.** Sellers submit documentation in diverse formats (PDF, images, scanned documents) with varying quality levels across multiple languages (English, Hindi, Arabic, Portuguese, Turkish). Validation requires extracting structured information from unstructured inputs and cross-referencing against eight distinct criteria: date verification, quantity reconciliation, product name matching through fuzzy logic, seller information cross-validation, tax registration validation, brand authorization verification, distributor authenticity confirmation, and regulatory compliance checking. Document quality varies significantly, with challenges including poor scan quality, handwritten text, mixed languages within single documents, and non-standard formats.

**Decision Complexity.** SOPs encode nuanced policy interpretations that vary by marketplace, product category, suppression type, and seller history. Supply chain verification alone requires tracing product provenance through multiple distributor levels, validating brand authorization chains, confirming date sequences across invoice hierarchies, and authenticating supplier credentials through external databases. Decision trees can involve 20+ conditional branches based on suppression reason codes, document availability, seller compliance history, and marketplace-specific regulations.

## B SOP Groomer Technical Implementation

### B.1 SOP Decomposition

We employ an LLM-based parser identifying procedural boundaries through linguistic markers ("then," "next," "if"), action verbs, and structural cues. The decomposition preserves hierarchical relationships between parent steps and sub-steps, enabling targeted refinement without losing procedural context. Each extracted step receives a unique identifier and metadata tags including step type, required inputs, expected outputs, and dependencies on previous steps. This structured representation enables precise targeting of ambiguities and facilitates automated validation of SOP completeness.

### B.2 Clarification Question Generation

Given a decomposed SOP, this component identifies ambiguous or underspecified instructions through pattern matching and semantic analysis. We employ RAG (Lewis et al., 2020) where the LLM receives: (1) the specific SOP step requiring clarification, (2) surrounding context from adjacent steps, and (3) relevant documentation retrieved from a knowledge base of platform guides, policy documents, and historical case resolutions. The system generates questions prioritized by their impact on automation feasibility, with navigation-blocking ambiguities (missing system names, undefined field references) addressed before optimization opportunities (alternative approaches, edge case handling).

### B.3 Multi-Modal Information Extraction

To minimize expert burden, we implement automated answer generation through three complementary extraction methods:

**Text Document Processing.** PDFs, wiki pages, and training materials are indexed using dense retrieval with domain-adapted embeddings. When a clarification question is generated, the system retrieves relevant passages that may contain implicit answers. For example, a question about "which system to check for suppression reasons" retrieves documentation mentioning the Account Health Dashboard and its specific fields.

**Browser Replay Analysis.** When experts demonstrate procedures through screen recordings, we extract step-by-step navigation sequences by analyzing DOM changes, click events, form submissions, and data extraction patterns. This captures tacit knowledge that experts may not explicitly document, such as the specific sequence of clicks needed to navigate nested menus or the visual cues used to identify correct data fields.

**Screenshot Understanding.** Interface screenshots are processed using vision-language models to identify UI elements, their spatial relationships, and associated labels. This enables automatic generation of element locators (XPath expressions, semantic descriptions) and navigation instructions that can be used by WebAgent for automation.

### B.4 Auto-Merge Integration

Generated answers are automatically integrated into the SOP at the precise location of the original ambiguity through template-based expansion. The merge process preserves document structure (headings, numbering, formatting) while expanding under-specified instructions into detailed, executable steps. A diff-based review interface highlights proposed changes, allowing domain experts to validate accuracy, modify phrasing, or reject clarifications that introduce errors. The iterative loop continues until all identified ambiguities are resolved and the SOP passes automated validation checks for completeness (all required inputs specified, all decision branches covered) and executability (all referenced systems and fields defined).

## C SOP Categories and Validation Criteria

Table 3 provides detailed information about the seven SOP categories evaluated in our experiments, including the primary document types, key validation criteria, and decision complexity for each category.

## D Document Validation Criteria Details

The Document Verification Agent validates seller-submitted documents against eight distinct criteria:

1. **Date Verification**: Invoice date must be within acceptable range (typically 90-180 days) and precede listing date.

2. **Quantity Reconciliation**: Quantities on invoice must match or exceed listed inventory quantities.

3. **Product Name Matching**: Fuzzy matching (>85% similarity) between invoice product descriptions and listing titles.

Table 3: SOP Categories with Document Types and Validation Criteria

| SOP Category | Document Types | Key Criteria | Decision Branches |
|---|---|---|---|
| Product Quality | Tax invoices, Images | Date, quantity, seller match, product description | 12-15 |
| Product Authenticity | LOA, Brand certificates | Brand authorization, distributor chain, dates | 18-22 |
| Trademark Infringement | Brand registration, LOA | Trademark validity, authorization scope | 15-18 |
| Copyright Violations | Licensing documents | Copyright ownership, usage rights | 10-12 |
| Counterfeit Claims | Supply chain invoices | Multi-level chain, brand linkage | 20-25 |
| Patent Design Rights | Patent certificates | Patent validity, design match | 8-10 |
| Wrong Item Sent | Delivery proof, Images | Item comparison, shipping records | 6-8 |

4. **Seller Information Cross-Validation**: Seller name/address on invoice must match registered seller details.

5. **Tax Registration Validation**: GST/VAT numbers validated against government databases.

6. **Brand Authorization Verification**: LOA validity, authorized product scope, and geographic coverage.

7. **Distributor Authenticity**: Verification of distributor credentials through external databases.

8. **Regulatory Compliance**: Product certifications match marketplace category requirements.

# E Reproducibility and Hyperparameters

## E.1 Language Model Configuration

All Agent-Ops components use a proprietary large language model as the underlying LLM. Table 4 summarizes the hyperparameters used across different agents.

Table 4: LLM Hyperparameters by Agent Component

| Component | Temp. | Top-p | Max Tokens |
|---|---|---|---|
| SOP Groomer | 0.3 | 0.9 | 4096 |
| WebAgent | 0.1 | 0.95 | 2048 |
| Document Verification | 0.2 | 0.9 | 4096 |
| Orchestrator | 0.2 | 0.9 | 2048 |

Lower temperature settings (0.1-0.3) were chosen to prioritize deterministic, consistent outputs critical for production reliability. WebAgent uses the lowest temperature (0.1) to ensure reproducible web navigation sequences.

## E.2 RAG Configuration

For SOP Groomer's retrieval-augmented generation:

- **Embedding Model**: A proprietary large-scale embedding model
- **Chunk Size**: 512 tokens with 50-token overlap
- **Top-k Retrieval**: k=5 for clarification questions
- **Similarity Threshold**: 0.75 cosine similarity

## E.3 Document Processing

- **OCR Engine**: AWS Textract for structured extraction
- **Fuzzy Matching**: Levenshtein distance with 85% similarity threshold
- **Multi-lingual Support**: Language detection via AWS Comprehend, with language-specific OCR models for Hindi, Arabic, Portuguese, and Turkish

## E.4 WebAgent Configuration

- **Browser**: Headless Chromium via Playwright
- **Action Timeout**: 30 seconds per element interaction
- **Retry Policy**: 3 attempts with exponential backoff
- **Screenshot Resolution**: 1920×1080 for visual grounding

## E.5 Computational Resources

Experiments were conducted on AWS infrastructure:

- **Orchestration**: AWS Lambda with 10GB memory
- **Average Latency**: 45 seconds per case (excluding document upload)
- **Cost**: Approximately $0.15 per case (LLM API calls)

### E.6 Evaluation Protocol

All reported metrics are averaged over 3 independent runs. Test cases were randomly sampled from production workloads with stratification by region, SOP category, and suppression type. Human expert annotations served as ground truth, with inter-annotator agreement of 94.3% (Cohen's $\kappa = 0.89$).

## F Error Analysis

Analysis of Agent-Ops failures across 500 test cases reveals the following error distribution:

- **Document Quality Issues (34%)**: Poor scan quality, handwritten text, or severely damaged documents that OCR cannot process reliably.
- **Ambiguous Policy Interpretation (28%)**: Edge cases where SOP decision rules are genuinely ambiguous even for human experts.
- **External API Failures (5%)**: Government portal timeouts or unavailability during tax registration validation.