# TOOL-AUGMENTED MULTIMODAL AGENTS FOR PIPING & INSTRUMENTATION DIAGRAM UNDERSTANDING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Piping and instrumentation diagrams (P&IDs) are key engineering documents that specify equipment, instrumentation, and piping connectivity in complex industrial systems. Despite their central role in engineering, procurement, and construction workflows, P&IDs remain largely interpreted manually due to dense layouts, heterogeneous drafting standards, and project-specific conventions. Interpreting these diagrams requires not only symbol detection but also reliable reasoning about line connectivity, topology, and semantic attributes—capabilities that existing approaches struggle to achieve. We introduce **PID-AGENTS**, the first tool-augmented multimodal agent framework for structured P&ID understanding. Rather than relying solely on static visual recognition, PID-AGENTS enables vision-language agents to actively explore diagrams through a set of novel *domain-specific drawing tools*, including region-of-interest rendering, event-driven line tracing, and local connectivity probing. Through iterative tool use, the agent gathers visual evidence and reconstructs pipeline-level relations and attributes beyond object detection. To support specification-driven reasoning, we further integrate *retrieval-augmented specification*, grounding attribute inference and consistency checks in external engineering documents. The resulting system converts rasterized P&IDs into connectivity-centric structured representations consisting of entities, typed relations, and provenance, enabling reliable end-to-end interpretation of real industrial drawings.

## 1 INTRODUCTION

Piping and instrumentation diagrams (P&IDs) are essential engineering documents that describe the connectivity of equipment, instruments, and piping within complex industrial systems. They play a central role throughout engineering, procurement, and construction processes (Toghraei, 2019). However, interpreting P&IDs remains largely a manual task due to dense layouts, heterogeneous drafting standards, and project-specific conventions. The resulting interpretation and verification workflows are time-consuming and error-prone, motivating the development of automated methods that can convert unstructured drawings into structured, machine-readable representations. Existing automated approaches (Mani et al., 2020; Han et al., 2025; Bisht et al., 2023; Oeing et al., 2022) often focus either on entity detection or on producing digitized diagrams without reliably inferring connectivity and semantics, limiting the development of scalable and reliable systems for holistic P&ID understanding. Effective interpretation requires not only accurate entity detection but also robust reasoning over line connectivity and semantic relationships among components.

P&ID understanding is challenging because these drawings *densely entangle heterogeneous modalities*. They contain small symbols, long and branching line networks, and spatially distributed text labels whose meanings depend on local context. Real industrial sheets often exhibit high levels of clutter, overlap, and visual similarity across symbol classes, while raster quality varies due to plotting and scanning artifacts as well as project-specific drafting conventions (Jamieson et al., 2025; Shteriyanov et al., 2025a). A particularly difficult aspect is *cross-project domain shift*. Symbol taxonomies, abbreviations, and graphical styles vary substantially across companies and even across projects, causing perception modules trained on one project to degrade on another (Han et al., 2025; Jamieson et al., 2024a). Moreover, annotation is expensive and often constrained by confidentiality,



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

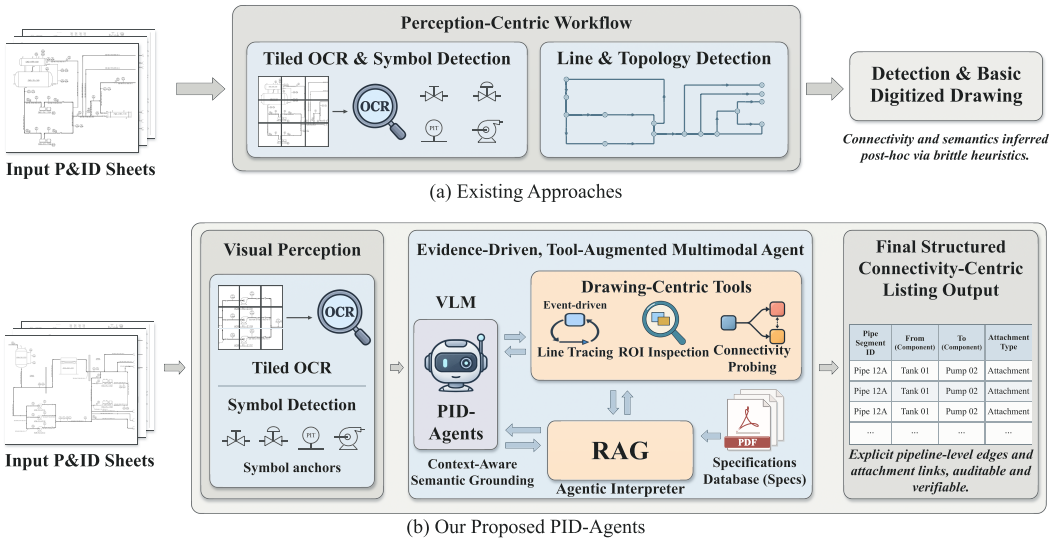



Figure 1: **Comparison of P&ID understanding pipelines. (a)** Existing approaches often terminate at entity detection, using post-hoc heuristics for connectivity. **(b)** Our PID-AGENTS framework incorporates *both* OCR and symbol detection with an evidence-driven, tool-augmented multimodal agent to iteratively reconstruct a structured connectivity listing grounded in project specs.

leading to *limited labeled data*, especially for rare symbols (Jamieson et al., 2024a; Bisht et al., 2023). These factors make P&ID interpretation a demanding instance of diagram understanding, where perceptual errors can easily cascade into structural mistakes.

Most prior work follows modular pipelines that combine optical character recognition (OCR), symbol detection, and line processing, followed by heuristic association or explicit graph construction. Early deep learning approaches combined convolutional neural network (CNN)-based symbol classifiers with OCR and graph traversal to detect line connectivity (Mani et al., 2020). Subsequent work explored unsupervised symbol localization to reduce annotation cost (Bisht et al., 2023) and graph neural networks for continuous line classification across projects (Han et al., 2025). More recently, transformer-based image-to-graph formulations have been proposed to jointly extract symbols and relations (Marius et al., 2025), and multimodal vision-language model (VLM)-based systems have incorporated legend information to enhance semantic interpretation (Shteriyanov et al., 2025b). However, none of these studies addresses the **complete understanding problem of P&ID drawings** beyond object detection. Practical deployment requires both perception and reasoning over entities, their connections, semantics, and project-specific specifications in order to produce structured outputs that are auditable and verifiable for downstream analysis. These requirements demand solutions that are connectivity-aware, robust to cross-project variation, and capable of providing interpretable evidence for their decisions.

In this paper, we propose **PID-AGENTS**, a P&ID listing pipeline that directly targets the full P&ID understanding problem by elevating connectivity to the primary objective. As in Figure 1, rather than stopping at symbol and text extraction, PID-AGENTS exports a structured, connectivity-centric representation (i.e., pipeline-level edges) that supports downstream validation and integration. To make this practical across real projects, our design is guided by two key properties: (1) *connectivity-centric outputs* that explicitly represent interconnections rather than inferring them post hoc, and (2) *evidence-grounded decisions* that improve auditability and reduce brittle heuristic dependencies.

To achieve this, we combine OCR and symbol detection with *evidence-driven* structural interpretation that iteratively gathers local observations before committing to connectivity and semantics. Concretely, PID-AGENTS rasterizes each sheet at high resolution, performs tiled OCR and symbol detection to propose candidate entities, and then uses a tool-augmented multimodal agent to reconstruct connectivity by querying drawing-centric tools (e.g., region-of-interest (ROI) inspection, event-driven line tracing, and local connectivity probing). To handle decisions conditioned on project documentation, we augment the interpretation process with retrieval over a project knowledge base so that specification-dependent attributes and consistency checks are grounded in explicit evidence. Our **contributions** are as follows.

- We formulate P&ID understanding as a connectivity-centric listing problem that explicitly reconstructs pipeline-level relations.
- We introduce an evidence-driven, tool-augmented multimodal agent for topology-aware connectivity reconstruction that replaces proximity-based heuristics alongside retrieval-based grounding to condition semantic interpretation on project specifications and constrain multimodal reasoning.
- We demonstrate the effectiveness of the proposed framework through a cross-project evaluation on real industrial P&IDs, showing gains in listing accuracy over VLM baselines.

## 2 PRELIMINARIES

Piping and instrumentation diagrams (P&IDs) encode pipelines, equipment, instruments, and control relationships throughout the engineering lifecycle. Detecting their individual symbols and text is insufficient for operational tasks such as material take-off, procurement, and installation planning, which also require connectivity, engineering conventions, and project-specific specifications. We therefore target a structured *listing*: records indexed by line numbers or instrument tags that consolidate entity types, connections, and specification attributes into data usable in engineering, procurement, and construction (EPC) workflows.

**Definition 1** (P&ID Listing)**.** *Let a P&ID document consist of a set of sheets $\{P_i\}_{i=1}^{N}$. Each sheet is rasterized into an image $I_i$. The P&ID listing problem is defined as the* ***task of extracting a structured collection of engineering records*** *$\mathcal{L}_i = \{\ell_{i,k}\}_{k=1}^{M_i}$, where each record $\ell_{i,k}$ corresponds to a pipeline or instrument entity identified in sheet $P_i$.*

Here, each record $\ell_{i,k} = (\text{identifier}, \text{entity type}, \text{connectivity}, \text{attributes})$, where *identifier* denotes the line number or tag label, *entity type* specifies the component category (e.g., pipe, valve, instrument), *connectivity* represents the attachment relationships between components, and *attributes* include specification fields derived from diagram text and project documentation. The goal is to recover $\mathcal{L}_i$ such that each record is geometrically grounded in the drawing evidence and semantically consistent with engineering specifications.

The task requires joint evidence from symbols, text, branching line networks, and project-dependent technical documents. Given $I_i$, the system identifies visual entities and their diagrammatic relations, gathers auxiliary observations through drawing-centric tools (ROI rendering, event-driven line tracing, and connectivity probing), and retrieves textual evidence from project knowledge bases. It must produce $\mathcal{L}_i$ consistent with drawing geometry, validated connectivity, and applicable specifications, while marking an attribute as unknown when the available evidence is insufficient. This formulation addresses the scale, cross-project variation, and multimodal ambiguity that make manual listing labor-intensive and error-prone.

## 3 METHODOLOGY: THE PID-AGENTS FRAMEWORK

The objective of PID-AGENTS is to convert rasterized P&ID drawings into structured engineering listings that describe pipeline and instrumentation information in a format directly usable for downstream engineering workflows.

### 3.1 FRAMEWORK OVERVIEW

As illustrated in Figure 2, the framework integrates perception modules, document-grounded reasoning, and parallelized agent-based interpretation to recover structured pipeline and instrument information from complex industrial diagrams. The system operates in two stages: (1) a global preprocessing and listing initialization stage, and (2) a parallel item-centric diagram interpretation stage. In **Stage 1** (§3.2), all sheets in a project are rasterized and analyzed using OCR and object detection models. These perception modules identify textual identifiers, such as line numbers and instrument tags, together with their spatial locations. The detected identifiers serve as anchors for subsequent reasoning. A language-model-based agent then reads project documentation (e.g., material specifications and guidelines) to infer specification attributes associated with each detected identifier. The extracted identifiers and inferred attributes are consolidated into an initial listing table that temporarily stores partially populated engineering records.

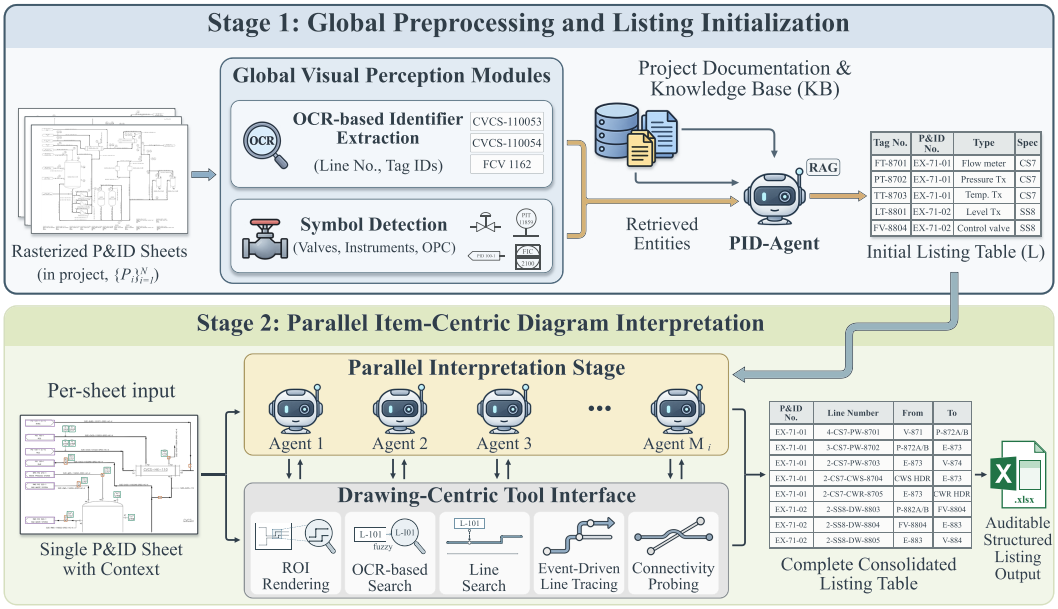



Figure 2: **Overview of PID-AGENTS.** In **Stage 1**, a project-wide preprocessing stage applies OCR and symbol detection to identify candidate line and instrument identifiers across all drawing sheets. PID-AGENTS then performs retrieval-augmented generation using the project knowledge base (e.g., material specifications) to infer and prefill a structured engineering listing. In **Stage 2**, each sheet is then parsed in parallel by multiple PID-AGENTS agents operating on the same drawing via a tool interface that provides drawing-centric utilities. Using these tools, the agents recover line connectivity and other attributes, producing the final structured P&ID listing.

Then, in **Stage 2** (§3.3), each P&ID sheet together with its detected entities is processed independently by a multimodal agent. Using the identifiers detected earlier as reference points, the agent performs localized reasoning with drawing-centric tools to recover connectivity information and additional attributes. Importantly, interpretation is conducted in an *item-based parallel manner*, i.e., each line number or instrument tag is processed independently with bounded runtime. The results are written back to the listing table until all entries are completed. The framework terminates when the listing table has been fully populated for all detected items.

### 3.2 STAGE 1: GLOBAL PREPROCESSING AND LISTING INITIALIZATION

The first stage of the framework performs OCR, symbol detection, and document-grounded specification prefilling. The objective of this stage is to rapidly identify all candidate engineering identifiers (line numbers and instrument tags) and initialize the structured listing table that will guide downstream interpretation.

**OCR for Engineering Identifiers** P&ID drawings exhibit substantial variability across projects due to differences in plotting pipelines, rendering styles, and engineering conventions. We train OCR models (Cui et al., 2025) on one project and evaluate them on another. Each P&ID sheet is rasterized at high resolution (e.g., $9925 \times 7017$ pixels) and processed using a tiled OCR strategy to avoid memory constraints associated with extremely large images. Pages are divided into overlapping tiles of $3200 \times 3200$ pixels with 1280-pixel overlap. OCR predictions produced in tile coordinates are mapped back to page coordinates using the tile offset, and duplicates from overlaps are consolidated using polygon-based non-maximum suppression, considering both spatial overlap and transcript similarity.

We also incorporate orientation-adaptive recognition for rotated text and apply data augmentation during training. Morphological dilation and erosion simulate variations in plotting thickness, while photometric transformations (e.g., compression artifacts, blur, noise) simulate scanning degradation.

**Symbol Detection** In addition to text identifiers, the system detects graphical symbols that represent instruments and key pipeline components. These detections provide contextual anchors for interpreting tag identifiers and validating local diagram semantics.

We train a one-stage object detector on a unified taxonomy of 48 symbol classes that are common across multiple industrial projects (Redmon et al., 2016). The taxonomy prioritizes common symbol categories such as valves, instrument markers, flow meters, and boundary indicators (e.g., tie-ins or off-page connectors). Equipment symbols are excluded because their shapes and annotations vary significantly across projects and are typically accompanied by textual identifiers that can be recognized via OCR. The detector outputs bounding boxes, class labels, and confidence scores, and is used to associate instrument tags with nearby symbols and identify connectivity boundaries that may serve as pipeline endpoints.

**Retrieval-Augmented Specification Grounding** While perception modules provide candidate identifiers, engineering specifications often depend on external project documentation, such as piping material specifications (PMS), insulation standards, and instrumentation design guidelines. To incorporate this contextual knowledge, we employ retrieval-augmented generation.

A project knowledge base is constructed by indexing engineering specification documents into structured information. The specification class and fluid code extracted from each pipeline identifier are used to retrieve the relevant document context. The retrieved specification fragments are provided as explicit evidence for attribute inference. Using these retrieved references, the agent extracts specification attributes associated with each identifier and records them in the temporary listing table. The agent records the evidence and assumptions underlying inferred attributes, leaving unresolved values as *unknown*.

### 3.3 Stage 2: Parallel Item-Centric Diagram Interpretation

After preprocessing, the system possesses a project-wide listing table containing detected identifiers and preliminary specification attributes. The second stage focuses on completing this listing by recovering connectivity information and additional diagram attributes. Each P&ID sheet, together with the bounding boxes of detected identifiers, is processed by a multimodal vision-language agent. Rather than performing monolithic page-level reasoning, the framework adopts an *item-centric execution strategy*. Each identifier (line number or instrument tag) defines an independent task that can be processed in parallel. This design significantly improves scalability for large industrial projects containing thousands of pipeline identifiers.

**Drawing-Centric Tool Interface** Here, we newly design a specialized tool interface that enables the agent to interact with several drawing-centric operations. The main tools are as follows.

- **ROI Rendering** tool extracts high-resolution local crops centered around identifiers or candidate endpoints.
- **OCR-Based Search** tool performs fuzzy matching to locate text identifiers across the drawing.
- **Line Search** tool presents numbered pipeline-segment candidates near a given seed point and initializes tracing from the selected segment.
- **Event-Driven Line Tracing** tool traverses binarized pipeline centerlines to follow pipeline paths.
- **Connectivity Probing** tool inspects nearby objects and identifies potential continuation points in the pipeline network.

These tools allow the agent to gather localized visual evidence and explore connectivity without relying solely on global pixel predictions.

Figure 3 illustrates a transition from directed tracing to exploration around its frontier. Fuzzy text matching and visual line selection supply the initial seed. The tracer advances rightward until it encounters a T-junction, where the next continuation requires a decision. A subsequent probe uses the returned junction coordinate as its input and explores the surrounding connected pixels, retrieving multiple contact candidates. The shared magenta marker links the trace frontier to the wider probing view. Returned contacts identify locations for further inspection; their semantic roles as pipeline endpoints still require interpretation.

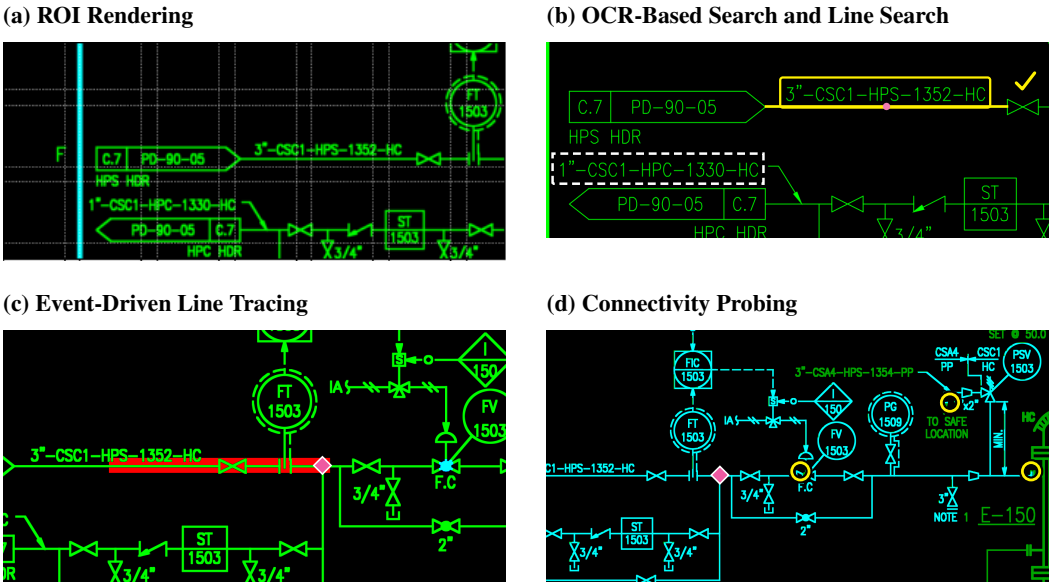



Figure 3: **Drawing-centric tool observations for pipeline interpretation.** For HPS-1352 in Project B, (a) ROI rendering exposes adjacent pipelines; (b) the selected OCR label and line are highlighted in yellow, and the alternative label is outlined in white; (c) the red trace stops at the T-junction marked by a magenta diamond; and (d) probing starts at that same junction, shown in a wider view shifted to the right. Cyan strokes show the recorded visited region, and yellow rings mark three returned contact candidates within this crop. These are controlled tool demonstrations, with (c)–(d) executed consecutively.

**Item-Based Connectivity Reconstruction** Connectivity inference is performed by tracing pipelines associated with detected line identifiers. Starting from the OCR location of a line number, the agent selects a nearby pipeline segment from the displayed candidates and traces the line in both directions. The tracer produces a stream of events corresponding to junctions, endpoints, or branching locations. At each frontier event, the agent renders a local ROI and probes nearby objects to identify potential terminal entities, e.g., off-page connectors, headers, or equipment connections. Candidate endpoints are ranked based on visual evidence, symbol detections, and contextual reasoning. The selected endpoints define the *from–to* connectivity record corresponding to each line identifier. As the interpretation is conducted independently for each identifier, PID-AGENTS can process multiple items concurrently. Each task is executed with bounded retries and a limited timeout threshold to ensure predictable runtime.

**Structured Listing Completion** The outputs produced during interpretation are written back to the project listing table, progressively filling missing fields, such as endpoints, component associations, and additional attributes derived from diagram evidence. The system continues processing identifiers until all listing entries have either been completed or reached the timeout limit.

For transparency and auditability, the framework optionally stores provenance information associated with each inferred attribute, including ROI coordinates and tool observations used during reasoning. This provenance enables efficient human verification and facilitates debugging when inconsistencies arise. The final output of PID-AGENTS is a structured engineering listing that aggregates information from all sheets in the project. The listing contains identifiers, connectivity information, and specification attributes, providing a structured representation suitable for downstream engineering analysis and EPC workflows.

## 4 EXPERIMENTS

We evaluate PID-AGENTS as an end-to-end system for automated P&ID listing under realistic *cross-project* deployment conditions. Our experiments focus on three questions.

- **(Q1)** How well does the proposed framework perform for end-to-end P&ID listing compared with strong VLM baselines?
- **(Q2)** How do OCR configurations perform in the cross-project evaluation?
- **(Q3)** How much do tool-based reasoning and document grounding contribute to recovering connectivity and specification attributes?

### 4.1 Experimental Setup

**Dataset and Cross-Project Evaluation** We evaluate the system under a realistic industrial setting where models must generalize across projects with different drawing conventions and raster characteristics. The framework is trained using data from *Project A* and evaluated on a disjoint *Project B*, which differ substantially in plotting configurations, symbol styles, and scan artifacts, creating a challenging cross-project domain shift. This dataset contains approximately 150 P&ID sheets obtained from a large petrochemical plant project, covering a wide range of process and utility systems and containing both piping and instrumentation information following international engineering standards (e.g., ASME and ISA).

For piping systems, the dataset contains approximately 3,500 pipeline instances, each with a line number, verified from–to connectivity endpoints, and engineering attributes such as design temperature and pressure. In addition, the dataset contains approximately 7,000 instrument instances, where each record includes a tag number, symbol location, and associated specifications such as service description, material of construction, connection type and size, and signal transmission type (e.g., 4–20 mA electronic signals). For quantitative evaluation, we annotate a subset of sheets with detailed ground truth for key listing fields, including 212 verified from–to pipeline connections, 622 instrument tag instances, 622 instrument-type annotations, and 622 line-connection attributes.

**Listing Tasks** To reflect real industrial workflows, we evaluate both *line listing* and *instrument listing* tasks. In the line listing task, the objective is to populate structured pipeline records using attributes typically required in engineering line lists, including the line identifier, `From`/`To` endpoints, and specification-related fields used in downstream EPC workflows (e.g., design temperature/pressure). Some attributes can be directly extracted from diagram text, while connectivity endpoints and connection attributes require interpreting diagram topology through line tracing and local connectivity reasoning. In the instrument listing task, the objective is to extract structured records for instrument components; each record includes the instrument `Tag No`, instrument type, associated symbol context, and additional attributes derived from project documentation.

**Comparison Baselines** Since no prior system performs complete end-to-end P&ID listing, we compare PID-Agents with a strong VLM baseline that directly interprets rasterized diagrams, along with variants augmented with perception outputs (OCR and symbol recognition) and with both perception outputs and retrieval-augmented generation (RAG) from project specifications. Specifically, all methods use *Gemini 3 Flash* as the backbone VLM capable of diagram reasoning. Additional specification retrieval is evaluated only for pipeline listing, with the same retrieval and generation procedure used by the RAG-augmented baseline and PID-Agents.

**Evaluation Metrics** We evaluate listing performance at the attribute level for each identifier. For each field, we report accuracy after applying simple string normalization (e.g., uppercasing and whitespace/punctuation canonicalization) to reduce spurious mismatches. Exact matching is used for most attributes, while instrument types and pipeline endpoints are evaluated using predefined partial-matching criteria. For connectivity-related fields such as `From` and `To`, we evaluate each endpoint field separately. We also report an overall average score computed as the mean accuracy across the evaluated fields in each task.

### 4.2 Implementation Details

Each P&ID sheet is rasterized at high resolution and processed with overlapping OCR tiles, after which text and symbol predictions are consolidated in page coordinates. These predictions initialize line and instrument records; project documents prefill specification fields, and the drawing-centric tools complete diagram-dependent connectivity fields. The resulting records are exported in JSON

Table 1: P&ID Listing results. We report field-wise accuracy for Pipeline Listing (top) and Instrument Listing (bottom); VLM uses Gemini 3 Flash.

| Method | Pipeline Listing | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Line No. | Fluid | Phase | Mat. | Rat. | PWHT | RT | Press. (Design) | Temp. (Design) | Test (Type) | Test (Press) | From | To | Avg. |
| VLM baseline | 0.9481 | 0.6085 | 0.5896 | 0.8396 | 0.0000 | 0.1887 | 0.0566 | 0.0849 | 0.2123 | 0.3349 | 0.0377 | 0.4623 | 0.4009 | 0.3665 |
| VLM + Perception | **0.9858** | 0.5425 | 0.5708 | 0.8396 | 0.0000 | 0.0425 | 0.0613 | 0.0802 | 0.2123 | 0.3160 | 0.0519 | 0.4387 | 0.4104 | 0.3501 |
| VLM + Perception + RAG | **0.9858** | **0.9858** | **0.8774** | **0.9245** | **0.9858** | **0.9858** | **0.9764** | **0.8538** | **0.8774** | **0.9717** | **0.7972** | 0.4575 | 0.4151 | 0.8534 |
| **PID-Agents** (Ours) | **0.9858** | **0.9858** | **0.8774** | **0.9245** | **0.9858** | **0.9858** | **0.9764** | 0.8491 | **0.8774** | **0.9717** | **0.7972** | **0.6321** | **0.6651** | **0.8857** |

| Method | Instrument Listing | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tag | Loc. From | Loc. To | Fail Position | I/O | Inst. Type | Fluid | Line Size | Line Spec | Avg. |
| VLM baseline | 0.6736 | 0.3360 | 0.3698 | 0.6704 | 0.5611 | 0.4180 | 0.1350 | 0.1945 | 0.2235 | 0.3980 |
| VLM + Perception | 0.7460 | 0.6672 | 0.6190 | 0.7444 | 0.5016 | 0.4357 | 0.2138 | 0.1318 | 0.2267 | 0.4762 |
| VLM + Perception + RAG | 0.7508 | 0.6768 | 0.6254 | 0.7492 | 0.5434 | 0.4277 | 0.2379 | 0.1495 | 0.2460 | 0.4896 |
| **PID-Agents** (Ours) | **0.9180** | **0.7878** | **0.8071** | **0.8441** | **0.7508** | **0.7846** | **0.7154** | **0.6559** | **0.7154** | **0.7755** |

and XML formats. Detailed image dimensions, parsing rules, retrieval procedure, and tool configuration are provided in §B.1. OCR and symbol detection results are reported in §4.4.

### 4.3 Main Results (Q1)

We evaluate the complete P&ID listing pipeline on *Project B* and compare it against the VLM baselines. Table 1 reports attribute-level performance across both line and instrument listing tasks.

Overall, PID-Agents consistently outperforms the baselines on fields that require diagram interpretation and connectivity reasoning. In particular, the framework achieves substantially higher accuracy on `From`/`To` connectivity fields and instrument tag extraction. These improvements stem from the integration of robust detection, symbol anchoring, and tool-based connectivity reconstruction. While VLM baselines can sometimes infer high-level diagram semantics, they struggle to reliably recover connectivity relationships in dense engineering drawings without explicit geometric reasoning. `From`/`To` endpoint identification remains the most challenging aspect of pipeline listing.

Specifically, Table 1 separates the contributions of perception, retrieval, and the agent stage. In line listing, retrieval raises the specification fields (e.g., rating from 0.00 to 0.99 and PWHT from 0.04 to 0.99) but moves From/To only from 0.44/0.41 to 0.46/0.42. The agent stage, with its drawing-centric tools, accounts for the From/To gains (0.46/0.42 to 0.63/0.67) and for the instrument fields that depend on linking a tag to its pipeline: Fluid, Line Size, and Line Spec rise from 0.24/0.15/0.25 to 0.72/0.66/0.72.

### 4.4 Additional Analyses

**OCR Performance (Q2)** Accurate OCR is essential for P&ID listing because line numbers and tag identifiers serve as anchors for all downstream reasoning tasks. Table 2 compares OCR configurations under the Project A → Project B evaluation. Fine-tuning improves recognition accuracy relative to the pretrained Pad-

Table 2: Cross-project OCR results on the private P&ID evaluation set (Project A → Project B). Recognition accuracy uses exact transcript matching.

| Method | Precision | Recall | F1-Score | Acc. (Recog.) |
|---|---|---|---|---|
| CRAFT + Tesseract | 87.75 | 81.07 | 84.28 | 56.77 |
| PaddleOCR | 66.94 | 75.57 | 70.99 | 88.62 |
| PaddleOCR (Fine-Tuned) | 63.16 | 53.44 | 57.90 | 98.33 |
| **PID-Agents** (Ours) | **93.47** | **92.44** | **92.95** | **98.50** |

dleOCR baseline, but reduces the F1 score in this evaluation. In contrast, the OCR configuration used in PID-AGENTS achieves the best performance with an F1 score of 92.95%. Furthermore, the number of detected text instances (10,515) closely matches the ground-truth count (10,611) in this evaluation. The symbol detector achieves an overall $AP_{50}$ of 84.00%; full per-class results are reported in Table 4.

**Effects of RAG- and Tool-Based Reasoning (Q3)** RAG supports specification attribute inference, while tool-based reasoning plays a critical role in recovering structured connectivity information that cannot be reliably inferred from pixel observations alone (Table 1). The drawing tools allow the agent to collect targeted visual evidence, trace pipelines, and inspect candidate endpoints. Without these tools, connectivity reconstruction becomes highly ambiguous in regions containing overlapping lines or dense annotations. The event-driven tracing mechanism enables the system to follow pipeline centerlines and identify branching events, while ROI inspection and connectivity probing allow the agent to validate semantic endpoints such as equipment ports or off-page connectors.

## 5 RELATED WORK

**P&ID digitization.** Digitizing legacy engineering drawings requires both perception of symbols, text, and linework and contextualization of their connectivity and drafting conventions (Moreno-García et al., 2019; Jamieson et al., 2024b). Early systems combined template matching, classical vision, OCR, and rule-based graph traversal (Kang et al., 2019; Rahul et al., 2019; Mani et al., 2020); more recent pipelines incorporate learned components (Kim et al., 2022). Although these methods establish strong entity-extraction pipelines, their topology recovery often remains tied to deterministic post-processing and project-specific assumptions, making it brittle under scan artifacts, gaps, dense overplotting, and cross-project variation.

**Symbol, text, and connectivity recognition.** Small symbols, visually similar classes, and long-tailed distributions have motivated semi-supervised and few-shot symbol detection as well as more realistic imbalanced benchmarks (Gupta et al., 2024; Jamieson et al., 2024a;c). Engineering OCR has likewise progressed from conventional text detection to learned detectors and integrated symbol–text systems (Yun et al., 2020; Liao et al., 2020; Zhou et al., 2017; Byun et al., 2025), while legend-conditioned VLMs improve semantic interpretation (Shteriyanov et al., 2025b). Connectivity recovery is commonly formulated through morphology, pixel-graph traversal, or line and arrow recognition (Mani et al., 2020; Bisht et al., 2023; Pillai et al., 2024; Moon et al., 2021; Han et al., 2025). Image-to-graph transformers instead learn relations directly (Shit et al., 2022; Marius et al., 2025), but require dense supervision and may not transfer readily across heterogeneous drawing styles. Our approach retains specialized perception while gathering local evidence interactively before committing to endpoints and relations.

**Vision-language models and tool-using agents.** VLMs support flexible multimodal and spatial reasoning (Bai et al., 2023; Qwen Team, 2025), but engineering benchmarks expose weaknesses under clutter, occlusion, and long-tail symbol distributions (Shteriyanov et al., 2025a). General tool-using agents combine language-model control with specialized modules (Shen et al., 2023; Yao et al., 2023; Yang et al., 2023; Surís et al., 2023; Gupta & Kembhavi, 2023), and retrieval-augmented generation grounds inference in external documents (Lewis et al., 2020). We instantiate these ideas for raster P&IDs by coupling OCR and symbol detection with drawing-centric tracing and connectivity tools, while retrieving project specifications for attribute inference.

## 6 CONCLUSIONS

This paper presents PID-AGENTS, a connectivity-aware framework for end-to-end P&ID listing, which turns rasterized P&IDs into line and instrument listings by combining OCR and symbol detection, retrieval-grounded specification inference, and a VLM agent with drawing-centric tools. On real industrial P&IDs (Project A $\rightarrow$ Project B), it raises From/To accuracy from 0.46/0.42 to 0.63/0.67 and the instrument-listing average from 0.49 to 0.78 over the strongest VLM baseline. The comparison also shows where each component helps. Retrieval recovers specification attributes but barely changes endpoints, whereas the endpoint gains come from the agent stage, which inspects and traces the drawing with tools. These results suggest that recovering connectivity benefits from acting on the drawing rather than reading it in one pass.

### AI USE STATEMENT

Generative AI tools, including OpenAI Codex, were used to assist with converting an existing manuscript between conference LaTeX formats, reorganizing sections, editing and condensing existing prose, integrating existing supplementary text as an appendix, and checking LaTeX structure, citations, cross-references, and presentation. They were not used during this format migration to generate experimental data or alter reported measurements. All AI-assisted changes in the submitted manuscript were reviewed by the authors, who take responsibility for the final content.

### REPRODUCIBILITY STATEMENT

The cross-project evaluation protocol, listing tasks, baselines, metrics, and implementation settings are described in Section 4.1. The perception and connectivity-reconstruction procedures are described in Sections 3.2 and 3.3, while additional OCR augmentation settings, the full symbol inventory, field-level failure analysis, and a qualitative execution trace are provided in the appendix.

## REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966. 9

Akhilesh Bisht, KR Chandrika, Divyasheel Sharma, and Deepti Maduskar. Pidpipe: A pipeline for digitizing p&id with unsupervised symbol localization. In *2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON)*, pp. 1–7. IEEE, 2023. 1, 2, 9

Junhyung Byun, Bonggu Kang, Duhwan Mun, Gwang Lee, and Hyungki Kim. Optimizing image format piping and instrumentation diagram recognition: Integrating symbol and text recognition with a single backbone architecture. *Journal of Computational Design and Engineering*, 12(6): 55–72, 2025. 9, 14, 20

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025. 4, 13, 14

Mohit Gupta, Chialing Wei, and Thomas Czerniawski. Semi-supervised symbol detection for piping and instrumentation drawings. *Automation in Construction*, 159:105260, 2024. 9, 13, 14

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962. IEEE, 2023. 9

Seung-Tae Han, Yoochan Moon, Ji-Beob Kim, Hyunsik Lee, and Duhwan Mun. Graph neural network-based method for classifying continuous lines in piping and instrumentation diagram. *Advanced Engineering Informatics*, 66:103457, 2025. 1, 2, 9, 16, 20

Laura Jamieson, Eyad Elyan, and Carlos Francisco Moreno-García. Few-shot symbol detection in engineering drawings. *Applied Artificial Intelligence*, 38(1):2406712, 2024a. 1, 2, 9, 13, 14

Laura Jamieson, Carlos Francisco Moreno-García, and Eyad Elyan. A review of deep learning methods for digitisation of complex documents and engineering diagrams: L. jamieson et al. *Artificial Intelligence Review*, 57(6):136, 2024b. 9, 16

Laura Jamieson, Carlos Francisco Moreno-García, and Eyad Elyan. A multiclass imbalanced dataset classification of symbols from piping and instrumentation diagrams. In *International Conference on Document Analysis and Recognition*, pp. 3–16. Springer, 2024c. 9

Laura Jamieson, Carlos Francisco Moreno-Garcia, and Eyad Elyan. Towards fully automated processing and analysis of construction diagrams: Ai-powered symbol detection: L. jamieson et al. *International Journal on Document Analysis and Recognition (IJDAR)*, 28(1):71–84, 2025. 1

Sung-O Kang, Eul-Bum Lee, and Hum-Kyung Baek. A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (p&id). *Energies*, 12(13):2593, 2019. 9

Byung Chul Kim, Hyungki Kim, Yoochan Moon, Gwang Lee, and Duhwan Mun. End-to-end digitization of image format piping and instrumentation diagrams at an industrially applicable level. *Journal of Computational Design and Engineering*, 9(4):1298–1326, 2022. 9

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020. 9, 17

Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11474–11481, 2020. 9, 13, 14

Shouvik Mani, Michael A Haddad, Dan Constantini, Willy Douhard, Qiwei Li, and Louis Poirier. Automatic digitization of engineering diagrams using deep learning and graph search. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 673–679. IEEE, 2020. 1, 2, 9, 16, 20

Stürmer Jan Marius, Graumann Marius, and Koch Tobias. From engineering diagrams to graphs: Digitizing p&ids with transformers. In *2025 IEEE 12th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–11. IEEE, 2025. 2, 9

Yoochan Moon, Jinwon Lee, Duhwan Mun, and Seungeun Lim. Deep learning-based method to recognize line objects and flow arrows from image-format piping and instrumentation diagrams for digitization. *Applied sciences*, 11(21):10054, 2021. 9

Carlos Francisco Moreno-García, Eyad Elyan, and Chrisina Jayne. New trends on digitisation of complex engineering drawings. *Neural computing and applications*, 31(6):1695–1712, 2019. 9

Jonas Oeing, Wolfgang Welscher, Niclas Krink, Lars Jansen, Fabian Henke, and Norbert Kockmann. Using artificial intelligence to support the drawing of piping and instrumentation diagrams using dexpi standard. *Digital Chemical Engineering*, 4:100038, 2022. 1

MR Pillai, LN Teja Tadepalli, SR Bhamre, Shreya M Gajbhiye, and Deepak Uplaonkar. Visualizing vitality: Decoding p&id connectivity via symbol detection and line analysis. In *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, pp. 1–8. IEEE, 2024. 9

Qwen Team. Qwen2.5 VL! Qwen2.5 VL! Qwen2.5 VL! Qwen Blog, 2025. URL https://qwenlm.github.io/blog/qwen2.5-vl/. Accessed: 2026-09-26. 9

Rohit Rahul, Shubham Paliwal, Monika Sharma, and Lovekesh Vig. Automatic information extraction from piping and instrumentation diagrams. *arXiv preprint arXiv:1901.11383*, 2019. 9

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. 5

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023. 9

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. In *European conference on computer vision*, pp. 422–439. Springer, 2022. 9

Vasil Shteriyanov, Rimma Dzhusupova, Jan Bosch, and Helena Holmström Olsson. Blueprintsymvl: A discriminative benchmark for vlm symbol recognition in engineering blueprints. *Results in Engineering*, pp. 108171, 2025a. 1, 9

Vasil Shteriyanov, Rimma Dzhusupova, Jan Bosch, and Helena Holmström Olsson. Enhancing ocr-based engineering diagram analysis by integrating diverse external legends with vlms. *Journal of Software: Evolution and Process*, 37(12):e70072, 2025b. 2, 9

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023. 9

Moe Toghraei. *Principles of P&ID Development*, chapter 5, pp. 45–68. John Wiley & Sons, Ltd, 2019. ISBN 9781119329503. doi: 10.1002/9781119329503.ch5. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119329503.ch5. 1

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023. 9

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. 9

Dong-Yeol Yun, Seung-Kwon Seo, Umer Zahid, and Chul-Jin Lee. Deep neural network for automatic image recognition of engineering diagrams. *Applied sciences*, 10(11):4005, 2020. 9

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2642–2651. IEEE, 2017. 9, 13, 14

# A Appendix Overview

This appendix is organized as follows.

- Appendix B summarizes implementation details that are useful for reproducing the submitted system, with emphasis on OCR augmentation and the full symbol inventory used for detection.
- Appendix C analyzes representative failure patterns across structured listing fields, emphasizing how OCR, topology reconstruction, document grounding, and cross-field dependency errors propagate to the final output.
- Appendix D presents two recorded execution traces on OPEN100, showing how the agent resolves *From / To* endpoints.
- Appendix E summarizes the current limitations of the framework.

# B Implementation Details

This section highlights two implementation details that are useful to make explicit in the appendix. The first is the OCR augmentation strategy, which is designed to improve robustness to rasterized engineering text under scale variation, small geometric perturbations, photometric changes, blur, noise, and partial occlusion. The second is the full 48-class detection label space, which is broader than the reduced set shown in the main paper for readability. These design choices are motivated by prior work showing that engineering-diagram OCR and symbol understanding are particularly sensitive to blur, scan artifacts, long-tailed categories, and project-specific style variation (Cui et al., 2025; Liao et al., 2020; Zhou et al., 2017; Gupta et al., 2024; Jamieson et al., 2024a).

## B.1 End-to-End Pipeline Configuration

Each P&ID PDF sheet is rasterized into a high-resolution image (e.g., $9925 \times 7017$). We run OCR on overlapping $3200 \times 3200$ tiles with 1280-pixel overlap and map tile-level predictions back to page coordinates using tile offsets, followed by consolidation to deduplicate overlaps, and we detect instrument-related symbols as spatial anchors for downstream reasoning.

From the consolidated OCR outputs, we form initial listing candidates for both pipelines and instruments. Line numbers are identified by matching OCR transcripts with a single regex pattern and parsing them into structured components (size, piping class/spec, fluid code, serial, and optional insulation), with common typographic variations normalized. Instrument tags are extracted by using detected symbol regions as anchors: OCR text within each symbol box is grouped by rows and concatenated into a tag string (e.g., `FT` + `1015A` $\rightarrow$ `FT-1015A`); fragments that do not form a valid multi-line tag are discarded.

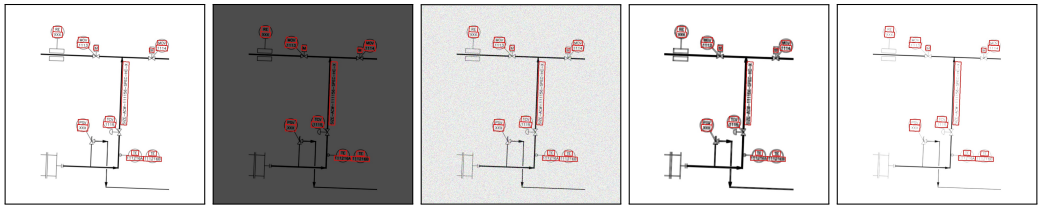
We perform document-grounded prefilling of specification fields using project documentation. PMS documents are converted into a structured lookup database and queried using the parsed line-spec and service/fluid code to prefill attributes such as material and rating, together with design/test-related fields when available; when evidence is missing, the corresponding attributes are left as *unknown*.

Finally, diagram-dependent fields are completed via a drawing-centric tool interface that exposes programmatic operations for agent-based interpretation. Given an identifier (line label or tag/symbol location), the interface returns high-resolution ROI renderings together with the OCR and symbol detections inside the ROI, supports fuzzy text search over page-level OCR for robust label localization, and provides utilities to snap query points to nearby pipe segments. The agent traces the pipe network bidirectionally and uses connected-object search to verify semantic endpoints (e.g., equipment connections or off-page connectors), thereby populating the `From`/`To` connectivity fields. For instrument listing, the agent associates each tag with its connected pipeline to populate tag–pipe attributes used in the instrument table. The completed line and instrument records are exported as structured tables (JSON and XML formats).

## B.2 OCR Augmentation

For OCR, the model must operate on text that is often small, low-contrast, and embedded in visually dense engineering drawings. To improve robustness, we apply appearance-level augmentations dur-

ing training, covering geometric perturbation, photometric variation, blur and noise, partial occlusion, and stroke-thickness variation. This is well aligned with recent OCR pipelines for both natural and engineering documents, where detector stability and recognition robustness depend strongly on augmentation under degraded raster conditions (Cui et al., 2025; Liao et al., 2020; Zhou et al., 2017; Byun et al., 2025). Figure 4 shows representative examples, and Table 3 summarizes the augmentations that directly modify the OCR input image. For clarity, we do not list preprocessing, target-generation, or tensor-conversion steps in the table.



(a) Original (b) Brightness (c) Gaussian (d) Thickening (e) Thinning

Figure 4: Examples of OCR augmentations used for training: original crop, brightness adjustment, Gaussian noise, stroke thickening, and stroke thinning. These transformations model appearance variations in rasterized engineering text.

Table 3: OCR augmentation configuration used during training to alleviate project-specific domain shift arising from heterogeneous document styles and image qualities. Only image-level augmentations that directly perturb the OCR input are reported.

| **Category** | **Augmentation** | **Configuration** |
|---|---|---|
| Geometric | Horizontal flip | probability $p = 0.50$ |
| | Rotation | angle range $[-5°, 5°]$ |
| | Random resize | scale range $[0.7, 2.0]$ |
| Photometric | Brightness / Contrast | brightness limit $[-0.25, 0.20]$, contrast limit $[-0.65, 0.35]$, $p = 0.45$ |
| | Gamma correction | gamma range $[65, 140]$, $p = 0.25$ |
| | CLAHE | clip limit 2.0, tile grid size $8 \times 8$, $p = 0.12$ |
| Blur / Noise | Gaussian blur | blur limit $[3, 7]$, $p = 0.22$ |
| | Gaussian noise | variance range $[10.0, 90.0]$, $p = 0.20$ |
| Occlusion | Random erasing | number of holes $[1, 10]$, erase size $[16, 80]$, fill value 255, $p = 0.12$ |
| Stroke variation | Stroke morphology | kernel size $[1, 2]$, thinning probability 0.65, iterations 1, $p = 0.25$ |

The table above is intentionally compact. Rather than listing many short rows with repeated prose descriptions, we group related transforms by function and report only the exact settings that are directly relevant to reproduction. This layout keeps the table readable while still exposing the main augmentation design choices.

### B.3 Full Symbol Inventory and Detection Results

The symbol detector is trained on a unified vocabulary of 48 classes obtained by consolidating the legend-sheet symbols from Project 1 and Project 2. In the main paper, however, we report per-class results only for the subset that is evaluable under the Project 1 $\rightarrow$ Project 2 setting. Classes that appear only in Project 1 are included in training but excluded from the main-paper per-class analysis because they do not occur in the Project 2 test set. This broader inventory also reflects the practical long-tail nature of symbol distributions in engineering drawings, where even infrequent categories can still matter as contextual anchors during downstream reasoning (Gupta et al., 2024; Jamieson et al., 2024a). Table 4 reports the per-class results, and Figure 5 provides a visual reference for the full detector label space.

Table 4: Per-class symbol-detection $AP_{50}$ results (in %) under the Project A → Project B protocol.

| Symbol | $AP_{50}$ | Symbol | $AP_{50}$ |
|---|---|---|---|
| All | 84.00 | Arrow | 97.10 |
| Ball Valve | 98.81 | Butterfly Valve | 99.50 |
| Check Valve | 99.50 | Concentric Reducer | 99.31 |
| Cylinder Actuator | 97.35 | DCS (Op. Access.) | 98.53 |
| Diaphragm Actuator | 91.61 | Diaphragm Seal | 62.84 |
| Eccentric Reducer | 77.27 | Generic Valve | 99.16 |
| Globe Valve | 99.50 | Locally Mounted Instr. | 98.69 |
| Open Drain/Sewer Conn. | 99.50 | Orifice | 99.50 |
| P&ID Cont. Drawing ID (Unit) | 49.08 | Relief Or Safety Valve | 99.50 |
| Self-Contained PR Reg. | 82.07 | Solenoid Valve | 99.50 |
| Spectacle Blind (Closed) | 94.11 | Spectacle Blind (Open) | 91.25 |
| Strainer Y Type | 99.50 | Tie-In Point | 99.50 |
| Var. Area Flow Meter | 70.68 | Vortex (Min. Straight Len.) | 99.50 |
| Work Scope | 99.50 | Logic Controller | 85.94 |
| Logic Ctrl. (Binary/Seq. PLC) | 93.67 | Magnetic | 75.90 |
| Strainer T Type | 99.50 | Battery Limit | 99.50 |

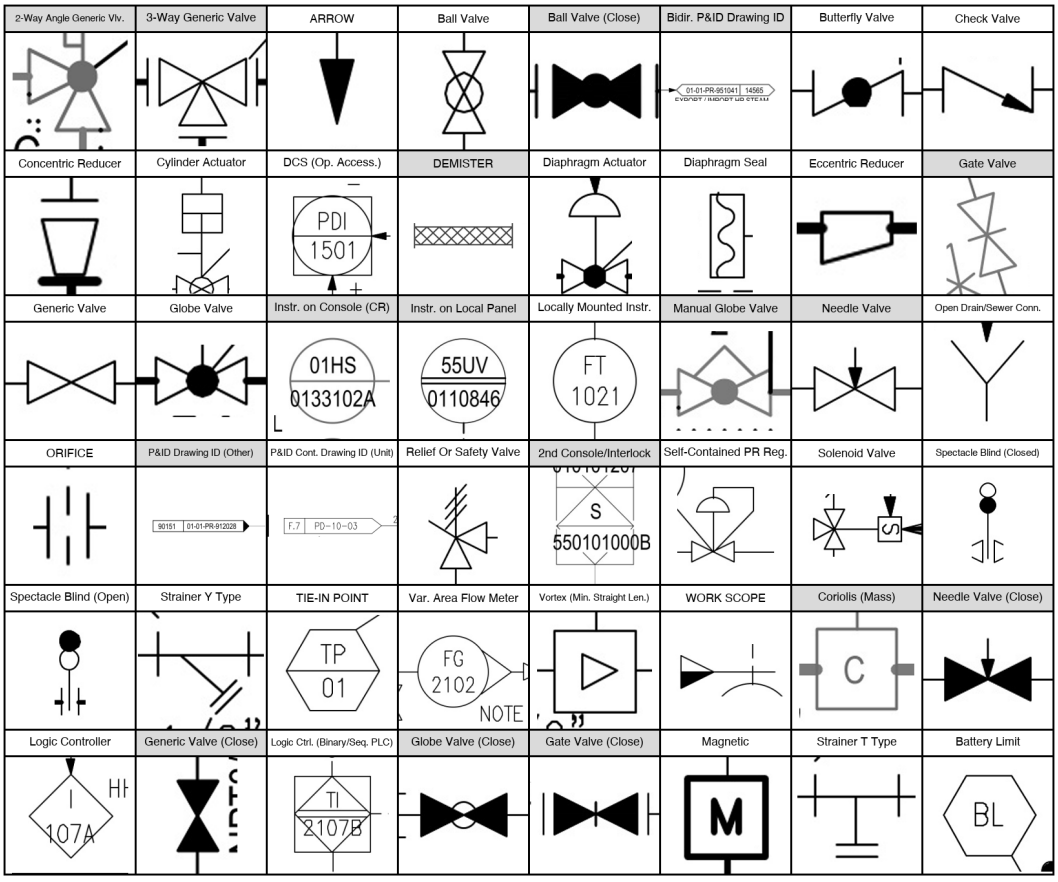



Figure 5: Full 48-class symbol inventory used for detection. The label space is constructed by consolidating legend-sheet symbols from Project 1 and Project 2. Gray-shaded entries denote classes observed only in Project 1 and therefore omitted from the main paper's per-class evaluation under the Project 1 → Project 2 protocol.

## C LISTING FIELD FAILURE ANALYSIS

This section focuses on why certain structured output fields fail and how those failures should be interpreted. Rather than treating every column as equally difficult, it is more informative to group fields by the kind of evidence they rely on: text-grounded identification, topology-dependent reasoning, document-grounded retrieval, and cross-field inheritance. Table 5 keeps a short description of each field for context, but the emphasis is on the dominant failure pattern observed for each column. This type of decomposition is also compatible with prior modular analyses of OCR, symbol detection, and line-connectivity recovery in engineering diagrams (Jamieson et al., 2024b; Mani et al., 2020; Han et al., 2025).

Table 5: Structured listing fields and representative failure patterns. The table keeps a compact field description for context while emphasizing the dominant source of error for each output column.

| Type | Field | Meaning | Typical Failure |
|---|---|---|---|
| Piping | Line No. | canonical pipeline identifier | OCR miss or duplicate-overlap consolidation error |
| | Fluid | fluid code parsed from the line identifier | identifier parsing error or incorrect fluid-code normalization |
| | Phase | phase or process-category field | ambiguous code semantics or missing mapped value |
| | Mat. Rat. | material / rating specification | PMS lookup miss or inconsistent line-spec parsing |
| | PWHT / RT | heat-treatment / inspection requirement | value not explicitly stated in retrieved specification fragments |
| | Design Press. / Temp. | design pressure / temperature | absent, scattered, or inconsistent document evidence |
| | Test Type / Press. | test type / test pressure | weak or missing explicit test-condition evidence |
| | From / To | semantic source / destination endpoints | wrong seed pipe, tracing break at cluttered junctions, arrow-direction error, or incorrect terminal equipment / connector selection |
| Instrument | Tag Loc. | OCR-recovered tag anchor location | fragmented OCR or weak symbol anchor near the tag |
| | Loc. From / To | local instrument connection endpoints | ambiguous local attachment, missed short branch, or failed local connectivity probing |
| | Fail Position | valve position in the event of failure | missed failure-state marking, confusion with normal position, or association with a neighboring valve |
| | I/O | signal or input-output attribute | tag normalization failure or ambiguous local signal notation |
| | INST TYPE | normalized instrument type | symbol-tag mismatch or visually similar symbol classes |
| | Fluid / Line Size / Line Spec | attributes inherited from the linked pipeline | incorrect linked pipeline or unresolved tag-to-pipe association |

A useful way to read the table is to separate failure sources into three broad families. First, text-anchored fields such as *Line No.*, *Fluid*, *Phase*, *Tag Loc.*, and *INST TYPE* usually fail because OCR fragments characters, merges neighboring tokens, or produces a normalized string that no longer matches the expected schema. These errors are often local, but they can trigger larger downstream mistakes once the wrong identifier is accepted as the anchor for subsequent reasoning. For *Fail Position*, drawing annotations and legend symbols must be associated with the correct valve or actuator; confusion with neighboring valves or normal open/closed designations can lead to incorrect assignments.

Second, topology-sensitive fields such as *From / To* and *Loc. From / To* fail for structural reasons. Here the system must localize the correct seed, snap it to the intended pipe or branch, trace connectivity through cluttered neighborhoods, and then choose the correct terminal candidate under local ambiguity. Missed short branches, broken centerlines, overlapping annotations, and visually subtle terminals can all perturb this chain, so these fields tend to accumulate multi-step errors rather than isolated OCR mistakes. This is consistent with earlier work showing that line extraction and connectivity reasoning remain brittle in the presence of clutter, branching ambiguity, and project-specific conventions (Mani et al., 2020; Han et al., 2025).

Third, document-grounded and inherited fields fail when the required evidence is weakly structured or depends on an earlier association. Fields such as *Mat. Rat.*, *PWHT / RT*, *Design Press. / Temp.*, and *Test Type / Press.* may be implicit, abbreviated, or scattered across retrieved project documents, making direct grounding unreliable even when the correct line or tag has already been recovered. Likewise, inherited fields such as *Fluid / Line Size / Line Spec* can collapse together once the upstream tag-to-pipeline linkage is wrong, producing correlated failures across several columns at once. In this sense, these columns expose the same evidence-selection bottleneck that retrieval-grounded systems more generally face when supporting attribute inference from partially structured documents (Lewis et al., 2020).

Overall, lower performance on a given field should therefore be interpreted in terms of the evidence pathway it requires. Text-centric columns are mainly bottlenecked by OCR and normalization; topology-centric columns remain limited by tracing and endpoint disambiguation; document-centric and inherited columns depend on whether the agent can recover sufficiently explicit evidence and maintain correct upstream associations.

## D CASE STUDIES: FROM / TO ENDPOINT IDENTIFICATION

These case studies illustrate how the agent combines geometric observations with equipment context and flow direction to identify *From / To* endpoints. Two pipelines from the OPEN100 Condensate System drawing illustrate equipment and off-page endpoint identification, respectively.

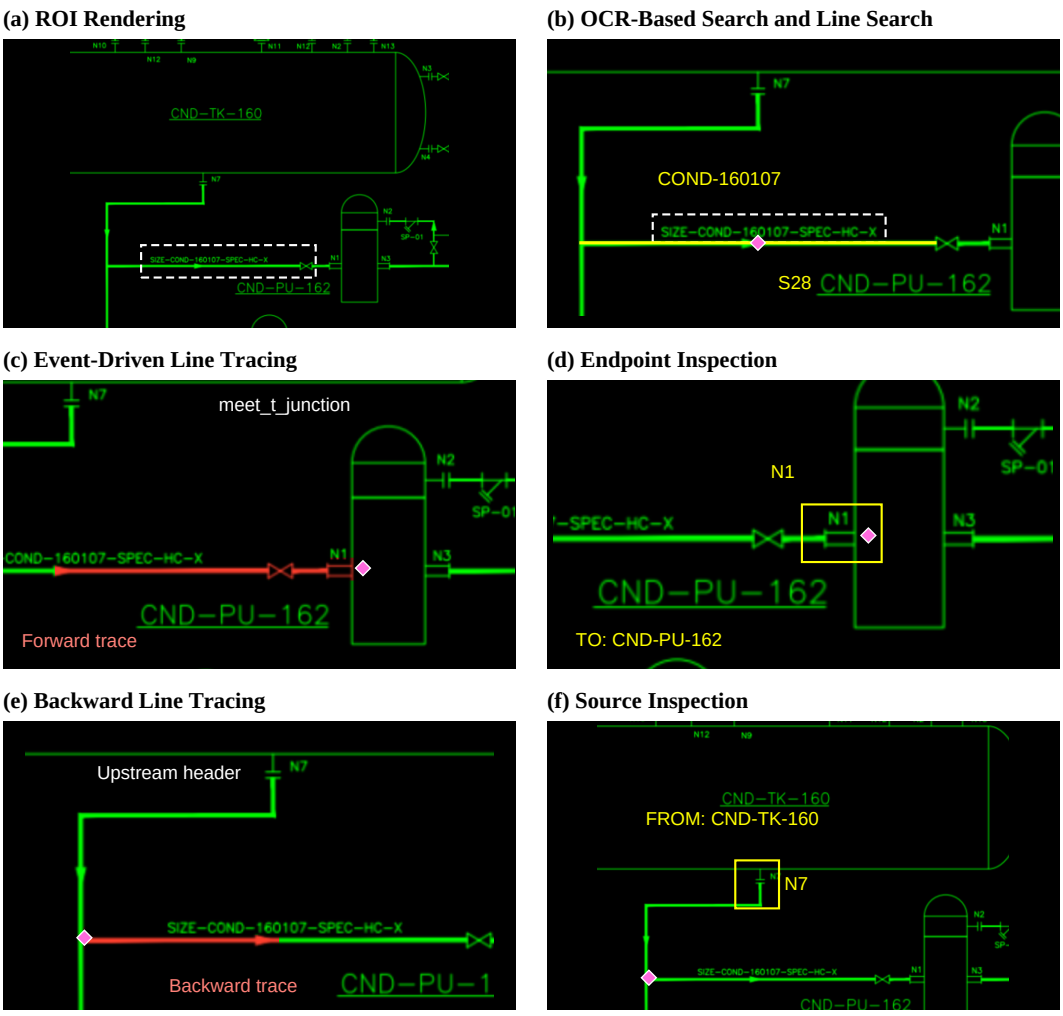



Figure 6: **Equipment endpoint identification for COND-160107.** (a–b) The agent localizes the target label and selects segment S28 through OCR-based search and Line Search. (c–d) Forward tracing reaches the pump outline; ROI inspection identifies inlet N1 of CND-PU-162. (e–f) Backward tracing reaches the supply header, and a wider view establishes its connection to tank nozzle N7. The resulting relation is `CND-TK-160` → `CND-PU-162`. White boxes mark search regions, red marks traced segments, magenta marks seeds or frontiers, and yellow marks selected segments and endpoints.

The tracer reports upward and downward continuations at the pump outline. Subsequent ROI inspection reveals inlet N1 and the equipment label, identifying CND-PU-162 as the destination. Backward tracing reaches the supply header; a wider view establishes the connection to CND-TK-160 through N7. This case illustrates how geometric events guide further inspection and how equipment context supports endpoint identification.

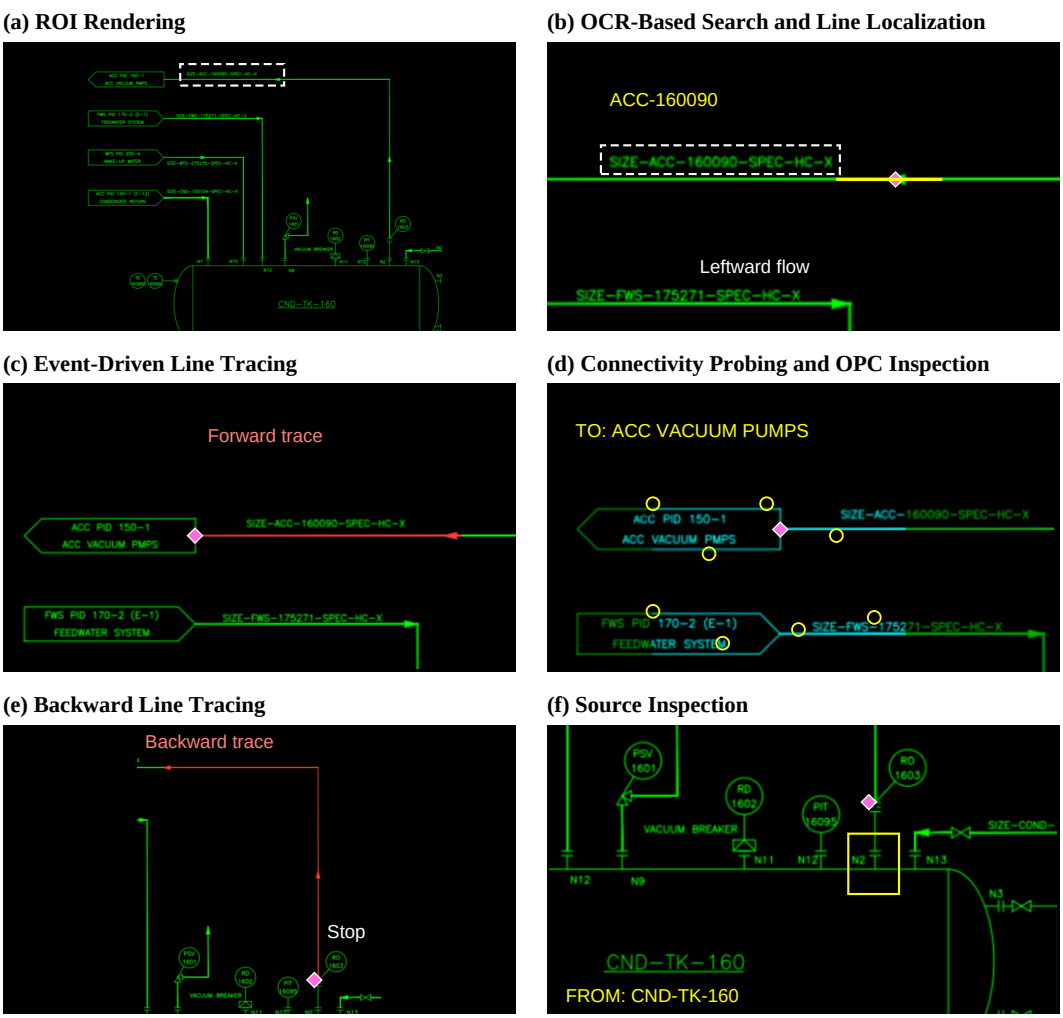



Figure 7: **Off-page endpoint identification for ACC-160090.** (a–b) ROI rendering and OCR-based search localize the pipe, and the leftward arrow establishes the forward tracing direction. (c–d) Tracing reaches the off-page connector (OPC); probing and local inspection support its identification as the ACC vacuum-pump connection. Cyan ink marks the probed region and yellow rings indicate returned contacts. (e–f) Backward tracing stops near RO-1603, followed by inspection of tank nozzle N2. The resulting relation is `CND-TK-160` → `ACC VACUUM PUMPS`, with the destination printed as `ACC VACUUM PMPS` in the drawing. Other colors follow Figure 6.

For ACC-160090, the horizontal and vertical arrows consistently indicate flow from the tank toward the OPC. Connectivity probing retrieves contacts on both the target line and a neighboring feedwater line. The selected vacuum-pump endpoint is supported by the connector's attachment to the traced line and its service label. Source assignment is completed by inspecting N2 after the backward trace stops near RO-1603. This example illustrates the complementary roles of flow cues, traced connectivity, and object context in *From / To* assignment.

## E LIMITATIONS

Despite encouraging results, several limitations remain. First, connectivity reconstruction can still degrade in regions with dense overplotting, broken centerlines, or strong local clutter, where multiple plausible continuations coexist within a small neighborhood. Second, semantic endpoint assignment remains difficult when terminal symbols are visually subtle, partially occluded, or highly project-specific. Third, severe raster artifacts, including scan noise, low contrast, and plotting degradation, can simultaneously weaken OCR, symbol anchoring, and line tracing. These challenges are broadly consistent with limitations already observed in prior engineering-diagram parsing and connectivity-recovery pipelines (Mani et al., 2020; Han et al., 2025; Byun et al., 2025).

Another practical limitation is that the current framework primarily reasons at the item level and does not yet impose stronger global consistency constraints across related pipelines and instruments. As a result, the system can still produce locally plausible but globally inconsistent partial outputs in difficult cases. Future work should therefore improve topology-aware global consistency, robustness to degraded scans, and uncertainty-aware fallback mechanisms for incomplete engineering evidence.