MASLOW: Multi-Agent Synthetic Data Generation for Underspecified, Low-Resource E-Commerce Classification

Ling Jiang[1,†], Shaobai Jiang[1,†], Ankur Gupta[1], Adalie Palma[1], Ali Marjaninejad[2], Xiaoyu Chu[1] and Prithvi Sen[1]

[1]*Amazon, Sunnyvale, CA, United States*
[2]*Amazon, Seattle, WA, United States*

Abstract

Training classifiers for e-commerce catalog management is bottlenecked by two simultaneous scarcities: underspecified task specifications (a bare URL or one-sentence description) and few or no labeled examples. We present MASLOW, to our knowledge the first multi-agent synthetic data pipeline that handles the full journey from underspecified inputs to labeled training data without requiring clean class labels, curated seeds, or task-specific corpora. Four specialized agents progressively enrich specifications, identify decision boundaries, generate diverse examples including hard negatives, and filter for quality. We evaluate MASLOW on 1,925 low-resource product-matching tasks and 40 unseen product classification tasks, two orders of magnitude beyond typical synthetic data studies. Ablation reveals that upstream context enrichment has outsized impact: removing the context-enrichment stage causes twice the performance drop (−5.8% F1) of removing any other agent, suggesting practitioners should prioritize input quality over generation architecture. Synthetic data complements human labels (+11.1% F1 on the scarcest tasks) and enables cold-start training from specifications alone. For In-Context Learning, task-aligned feedback (MASLOW+AFB) that regenerates examples receiving uncertain downstream predictions recovers quality that naive feedback loses. At 47× lower cost and 31× lower latency than human annotation, MASLOW processes 1,925 tasks in approximately one hour.

Keywords

synthetic data generation, multi-agent systems, low-resource learning, e-commerce classification, large language models

1. Introduction

Large e-commerce stores operate a large number of product classifiers spanning use cases from product type categorization to tax classification, safety compliance, and recall enforcement. Each classifier must distinguish target products from a large-scale product catalog, with the continued expansion of catalog coverage driving a high volume of new classification tasks annually. New tasks routinely arrive in a *low-resource* setting characterized by two simultaneous scarcities: fewer than 20 labeled examples (or none at all), and underspecified **task specifications** (natural-language descriptions of what a classifier should detect) that may be as minimal as a regulatory URL, a one-sentence requirement, or informal notes scattered across multiple sources (Figure 1).

Product recalls exemplify this challenge: when a regulatory agency issues a recall notice, it describes which products are affected but contains no identifiers linking to specific products in the catalog. The recall may affect only a small fraction of products in the catalog, and at announcement time there are typically zero labeled examples. Similar conditions arise for new regulations, emerging fraud patterns, and changing compliance requirements. In all cases, traditional approaches require collecting and annotating examples before training can begin, delaying enforcement. Specification quality presents an additional challenge: requirements arrive from diverse non-technical sources, each providing only partial information, while the knowledge needed for precise classification is scattered across external

GenAIECommerce'26: The Third Workshop on Agentic and Generative AI for E-Commerce, co-located with RecSys, September 28, 2026, Minneapolis, MN, USA

[†]These authors contributed equally.

jiangll@amazon.com (L. Jiang); shaobaij@amazon.com (S. Jiang); ankurgp@amazon.com (A. Gupta); adaliep@amazon.com (A. Palma); alimarj@amazon.com (A. Marjaninejad); xiaoychu@amazon.com (X. Chu); prithsen@amazon.com (P. Sen)

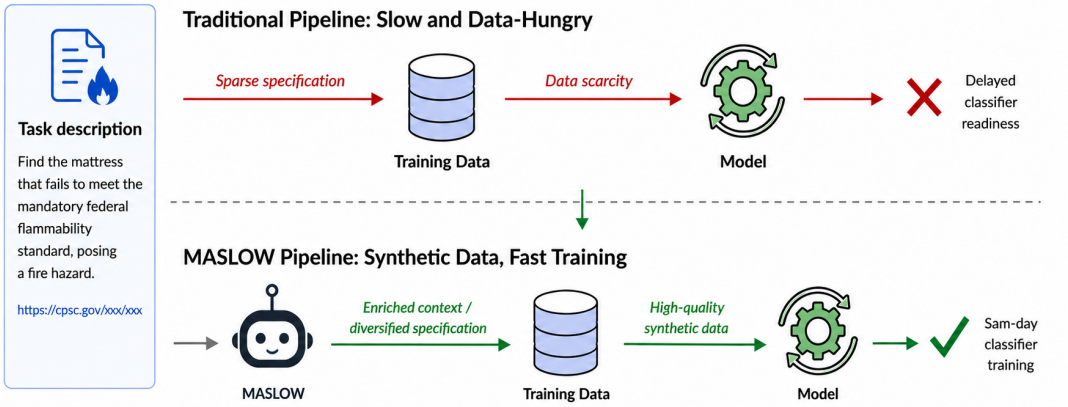


Figure 1: MASLOW addresses two key bottlenecks in low-resource e-commerce classification: sparse task specifications and inadequate labeled data. By enriching specifications, identifying decision boundaries, and generating high-quality synthetic data, MASLOW enables same-day classifier training without extensive manual annotation.

documents, catalog metadata, and domain expertise. At the volume of new tasks arriving annually, manually consolidating these scattered sources into a coherent specification for each task is impractical.

Synthetic data generation has advanced rapidly, from instruction-following synthesis [1, 2] to multi-agent pipelines with validation and iterative refinement [3, 4, 5]. However, key challenges persist: producing examples near decision boundaries is difficult with underspecified contexts [6, 5], and ensuring label correctness and diversity is nontrivial for long-tail tasks [7, 8]. Critically, existing frameworks assume access to structured task definitions, curated seed examples, or task-specific corpora [3, 4]. Industrial e-commerce settings seldom provide these; practitioners often receive only a bare URL or one-sentence description (Figure 1), with no seed examples. Prior approaches assume well-defined starting points: they either generate free-form text from predefined category labels (e.g., "sentiment: positive") [9, 2] or synthesize structured records from existing data distributions. None addresses generating complete labeled training examples from natural-language specifications alone. Our setting is fundamentally different: the input is an opaque URL (e.g., a government recall notice) that must first be fetched, parsed, and decomposed into discriminative criteria before any structured training data can be generated, and the target products may not yet exist in any dataset (Section 3). These gaps between existing pipelines and real-world industrial conditions motivate our work.

We present MASLOW, a multi-agent system that solves the *specification-to-classifier cold-start problem*: given only an underspecified task description (a URL, a sentence, or scattered notes), MASLOW generates complete synthetic product representations from scratch: structured, multi-attribute entries (title, brand, product type, features) paired with binary labels, sufficient to train a working classifier without any human-labeled examples. Four specialized agents progressively enrich context (*Extractor*), identify decision boundaries (*Analyzer*), generate diverse examples including hard negatives, samples that closely resemble target products but should not match (*Generator*), and filter for quality (*Validator*).

We empirically validate this design by comparing against single-agent monolithic baselines on 1,925 low-resource product-matching tasks using supervised learning, two orders of magnitude beyond the 5–15 datasets typical in synthetic data studies [3, 4], and on 40 unseen product classification tasks using few-shot In-Context Learning (ICL), where training-free inference allows rapid iteration on example quality. Our experiments reveal that upstream specification enrichment is the highest-leverage component rather than the generation architecture itself, and that distributional alignment with human data, not raw diversity, predicts downstream performance. For ICL, task-aligned feedback (MASLOW+AFB) that uses downstream classifier signals recovers the quality that naive feedback loses.

This paper makes the following contributions:

- **From underspecified specifications to classifiers.** MASLOW is, to our knowledge, the first multi-agent synthetic data pipeline with specification enrichment as an explicit stage, handling the full journey from underspecified input to labeled synthetic data without requiring clean class labels, curated seeds, or task-specific corpora.
- **Seed-free generation via Extractor and Analyzer.** The Extractor enriches sparse inputs into structured specifications; the Analyzer decomposes these into discriminative criteria and decision boundaries, enabling three-tier sample generation (positives, hard negatives, easy negatives) without any labeled seed examples.
- **Comprehensive evaluation across settings and scales.** Across 1,925 product-matching and 40 unseen product classification tasks, we demonstrate cold-start training, +11.1% F1 complement to human labels, and 47× cost reduction, while identifying that naive feedback degrades quality by disrupting label balance.
- **Synthetic demonstrations for In-Context Learning.** A large synthetic pool enables retrieval-based demonstration selection that substitutes for and complements scarce human examples. ICL's training-free inference further enables task-aligned feedback (MASLOW+AFB), which runs synthetic examples through the downstream classifier and regenerates those receiving uncertain predictions, revealing that feedback must align with the downstream task to be effective.

2. Related Work

Synthetic data generation. A growing body of work uses LLMs to generate labeled training data for downstream classifiers, including self-guided refinement [2], targeted generation from model error patterns [5], and class-conditioned prompting from label names [9]. Multi-agent approaches further decompose generation: MetaSynth [3] orchestrates expert agents for diverse pre-training corpora, GRAID [4] combines geometric constraints with multi-agentic reflection, and Gen-n-Val [10] applies agentic generation and validation for image data. Studies on synthetic data quality find that performance degrades on nuanced tasks with subtle decision boundaries [6] and that returns diminish beyond certain volumes [8]. A common assumption across this body of work is that the task is already well-defined: either by clean class labels, structured error analysis, curated seed examples, or task-specific corpora. MASLOW relaxes this assumption by including specification enrichment as an explicit stage, addressing settings where even the task definition is ambiguous.

Synthetic data in e-commerce. Synthetic data has been applied to product review generation [11], catalog enrichment [12], and social media content generation [13]. These focus on content expansion for existing catalog entries rather than generating task-aligned labeled datasets for classifier training from underspecified inputs.

Hard negative generation. Hard negatives that are semantically close to positives produce the strongest gradient signal in contrastive learning [14], and Josifoski et al. [15] generate challenging synthetic examples for information extraction. However, both lines of work require *existing data distributions* to mine or generate hard negatives. In our setting, hard negatives must be produced from specifications alone without any seed examples.

Low-resource and zero-shot classification. Few-shot learning [16] and ICL address label scarcity, with Liu et al. [17] showing example relevance dominates quantity. PECOS [18] enables cross-task transfer for extreme multi-label classification under scarcity. Meng et al. [9] generate training data from class names alone, but their tasks are defined by unambiguous labels with well-understood semantics. In our setting, even the class definition must be extracted and analyzed before generation can occur.

3. MASLOW: Multi-Agent Architecture

MASLOW processes noisy, incomplete task descriptions to generate task-specific synthetic product data through four sequential stages (Figure 2): extraction, analysis, generation, and validation. Agents

communicate through structured schemas capturing generation guidelines, synthetic samples, and validation scores. We illustrate each agent using a product recall as a running example.

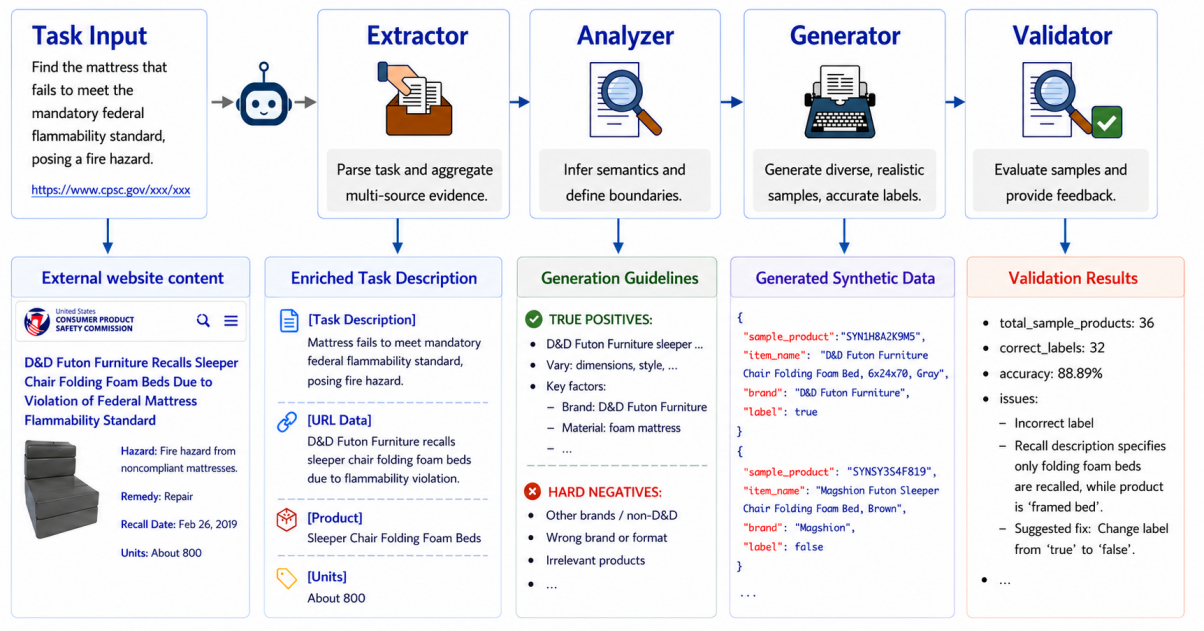


Figure 2: MASLOW comprises four specialized agents: Extractor aggregates multi-source information; Analyzer infers task semantics and identifies decision boundaries; Generator produces diverse synthetic examples; Validator evaluates quality.

Extractor addresses the underspecification problem directly by gathering and aggregating information from heterogeneous sources including regulatory webpages, internal documents, and informal communications, transforming partial or ambiguous inputs into structured, actionable formats (Figure 2; Appendix A). This fills domain knowledge gaps that would otherwise propagate through all downstream agents. It automatically identifies URLs via LLM prompting (Appendix C.1), fetches webpage content via crawler, and identifies example catalog items for few-shot demonstrations when available. The Extractor's output is a structured specification containing the original task description, extracted content from external sources, identified product information, and any metadata (e.g., affected units, dates) relevant to classification criteria.

Analyzer synthesizes extracted information to infer task semantics and identify critical decision boundaries. It decomposes complex requirements into clear decision points by addressing: "What attributes distinguish target items from similar products?" and "What product variations might incorrectly match but should be excluded?" Concretely, the Analyzer produces structured generation guidelines containing: (1) positive criteria defining what makes a product match the specification, with specific product types and examples; (2) hard negative criteria specifying near-miss exclusions; (3) discriminative attributes (e.g., brand, material, dimensions) that determine label membership; and (4) key characteristics to vary across generated samples. In our running example (Figure 2), the Analyzer identifies brand ("D&D Futon Furniture") as the primary discriminative attribute and material (foam mattress) as a critical secondary identifier. This captures a subtle distinction: only this specific manufacturer's products are recalled, despite functionally similar products from other brands existing in the catalog. When few-shot examples are available, the Analyzer uses them to refine its guidelines, paying particular attention to any human-corrected labels that reveal non-obvious decision boundaries. This structured output bridges the gap between the Extractor's enriched-but-unstructured specification and the precise instructions the Generator needs to produce label-accurate data without seed examples. Figure 3 shows a simplified version of the Analyzer prompt.

Generator leverages guidelines from the Analyzer to produce diverse, realistic synthetic products with binary labels (Figure 2; Appendix C.2). The Generator employs temperature sampling (T=1.0) to

```
[System]
You are an expert product analyst. Analyze task specifications
and generate product categories for synthetic data generation.
Focus on identifying key characteristics that define matching
vs non-matching products.

[User]
Analyze the following task specification and generate categories.
<task_specification> {TASK_SPECIFICATION} </task_specification>
<few_shot_examples> {FEW_SHOT_EXAMPLES} </few_shot_examples>

Based on the specification, generate:
1. TRUE POSITIVES: Products that SHOULD match
2. HARD NEGATIVES: Similar products that should NOT match
For each category, provide:
- Clear description of what products belong
- Specific product types/examples
- Key characteristics to vary
- Reasoning for category membership
```

Figure 3: Simplified Analyzer prompt. The Analyzer receives the enriched specification and produces structured generation guidelines defining positive and hard negative categories with discriminative attributes.

balance diversity with label consistency. Generation prompts specify output format, instruct variation of non-critical attributes while preserving label-determinant characteristics, and include few-shot examples when available. In our running example (Figure 2), the Generator creates positive examples (products that should be recalled) emphasizing critical brand and material attributes, and hard negative examples (products that should not be recalled) that test decision boundaries. Hard negatives differ in only one or few critical attributes, unlike easy negatives that differ in many ways (e.g., a toy vs. a recalled futon). For instance, a hard negative may match the recalled product in dimensions, color, and product type but differ only in brand (see Appendix B). This strategy is grounded in the principle that near-boundary examples sharpen classification margins: in contrastive and metric learning, hard negatives that are semantically close but belong to different classes produce the strongest gradient signal and the tightest decision boundaries [14]. By generating both target-matching positives and boundary-testing hard negatives, the Generator improves robustness against easily confused cases.

Validator evaluates generated samples across three dimensions: (1) label correctness relative to the Analyzer's decision boundaries, (2) realism and consistency of product attributes, and (3) diversity and edge case coverage (Figure 2; Appendix C.3). In our running example, 32 of 36 samples pass validation (88.9% pass rate); this is an internal quality filter, not a downstream classification metric. Problematic samples receive detailed rejection reasons. The Validator operates in two modes: *passive filtering*, where flagged samples are removed before training, and *iterative feedback* (MASLOW+FB, FeedBack), where the Validator's rejection reasons are returned to the Generator, which analyzes the flagged patterns and regenerates a fresh batch to avoid repeating them (Appendix C.2). We evaluate both modes in Section 4.

We implement our prototype using Claude Opus [19]. Each agent operates through engineered prompts; generation cost and latency are reported in Section 6.

4. Experiments: Supervised Learning

We evaluate our prototype on an offline large-scale low-resource product-matching benchmark, which requires rapidly identifying products matching a sparse specification with zero or few labels. Label scarcity is a defining characteristic of this domain: across many product-matching experimental tasks created, 32% had any positive labels, and only 6% had more than ten positive examples months after creation. While negative examples are abundant in the catalog, positive label scarcity is the bottleneck.

This represents an extreme multi-label classification challenge: diverse tasks with severe class imbalance where products can match multiple specifications simultaneously. In practice, such classifiers serve as a candidate generation stage that narrows a large catalog to a manageable review set for human verification. Absolute performance metrics are therefore inherently modest compared to traditional binary classification; the value lies in reducing the search space by orders of magnitude.

4.1. Experimental Setup

Dataset. We collected 1,925 product-matching tasks with human labels sufficient for held-out evaluation. Positive labels are the bottleneck: negative examples are abundant in the large catalog while only a small fraction of products match each specification.

Synthetic Data Generation. Using MASLOW, we generated approximately 400 synthetic products per task and retained 200 (100 positive, 100 negative) after Validator filtering, in zero-shot mode (no human examples in the generation prompt). We chose 200 as performance gains diminish sharply beyond 150 samples (Appendix D). We also evaluate few-shot generation, where up to 20 human labels are provided as seed demonstrations to the Generator.

Classification Model. With many tasks having fewer than 10 positive training examples, individual binary classifiers would severely underfit. We employ PECOS [18], a multi-label extreme classification framework that learns implicit relationships across tasks, enabling low-resource tasks to benefit from patterns learned in better-resourced ones. The multi-label formulation is also necessary because individual products can be affected by multiple specifications simultaneously.

Baselines. Most existing studies on multi-agent synthetic data generation assume structured inputs unavailable in our setting. For example, MetaSynth [3] requires domain-level descriptions, while GRAID [4] requires labeled seed examples and fine-tunes a local model. Neither accepts a bare URL or one-sentence specification as input. We instead compare against *single-agent* baselines that combine analysis, generation, and validation in a single monolithic prompt, generating in the same batch size ($\sim$50 products per LLM call): (1) **SA_raw** uses the original task description without enrichment, and (2) **SA** receives the same enriched context from the Extractor. This isolates the contribution of multi-agent decomposition from context enrichment. We also evaluate **MASLOW+FB**, which adds an iterative feedback loop where the Validator's rejections are fed back to the Generator for re-generation. Note that SA is a cost-lower single-prompt baseline and MASLOW+FB constitutes a self-refinement baseline; a fully cost-matched multi-step baseline is left to future work.

Evaluation Design. We denote training configurations as *Nh+Ns* (e.g., 20h+200s = 20 human-labeled + 200 synthetic examples per task). We evaluate three scenarios:

- **Cold-start** (0h+200s): train with 200 synthetic examples only. Tests whether MASLOW can train a working classifier from task specifications alone on day zero.
- **Complement** (20h+200s): augment a fixed budget of up to 20 human labels with 200 synthetic examples. Tests whether synthetic data improves over human-only training.
- **Budget allocation** (100 total, varying ratio): fixes total training size at 100 samples per task and varies the human-to-synthetic ratio (100h+0s through 0h+100s), providing a volume-controlled comparison that reveals the per-sample value of each data source.

We report results on 1,925 tasks with sufficient labels for evaluation. We stratify by positive label count into nested strata: =1 (786 tasks), <5 (1,277 tasks), and all (1,925 tasks). We report micro precision, recall, and F1, which reflect operational performance and are stable under the class imbalance present in this benchmark.

4.2. Cold-Start and Complement Results

MASLOW enables cold-start training and complements human labels. Without any human labels, MASLOW achieves F1=0.331 on the most scarce tasks (=1 label) and 0.246 overall, consistent with a candidate generation role that narrows the review space on day zero. When human labels become

Table 1
PECOS performance in cold-start (0 human labels) and complement (20 human + 200 synthetic) settings (micro P/R/F1, 1,925 tasks). Columns stratify by positive label count. SA = single-agent baseline; SA_raw omits context enrichment. MASLOW+FB adds iterative feedback. Best per setting and stratum in **bold**.

Pipeline	Tasks =1 label (786)			Tasks <5 labels (1,277)			All tasks (1,925)		
	P	R	F1	P	R	F1	P	R	F1
Cold-start (0h+200s)									
SA_raw	0.535	0.183	0.273	0.540	0.199	0.291	0.544	0.121	0.198
SA	0.578	0.226	0.325	0.566	0.230	0.327	**0.624**	0.132	0.218
MASLOW	**0.590**	**0.230**	**0.331**	**0.581**	**0.234**	**0.334**	0.608	**0.154**	**0.246**
MASLOW+FB	0.566	0.212	0.309	0.556	0.227	0.323	0.606	0.136	0.223
Complement (20h+200s)									
Human-only (20h)	**0.758**	0.327	0.457	**0.750**	0.515	0.610	**0.686**	0.359	0.471
SA_raw	0.662	0.447	0.533	0.693	0.597	0.641	0.647	0.405	0.498
SA	0.664	0.464	0.546	0.696	0.609	0.650	0.665	0.433	0.525
MASLOW	0.686	**0.485**	**0.568**	0.710	**0.619**	**0.661**	0.663	**0.440**	**0.529**
MASLOW+FB	0.660	0.471	0.549	0.694	0.612	0.650	0.667	0.435	0.527

available, performance improves substantially: MASLOW with 20 human labels achieves F1=0.568 on =1 tasks (+11.1% over human-only), driven primarily by recall gains (0.327→0.485) with a modest precision trade-off (0.758→0.686), indicating synthetic data helps the model identify target products it would otherwise miss. On the 786 single-positive tasks, MASLOW improves Mean Reciprocal Rank (MRR) from 0.317 (human-only) to 0.465, surfacing the target product at rank 2 on average with Recall@1 of 44.7% (Appendix E).

MASLOW outperforms monolithic baselines, especially in cold-start. Compared to SA_raw, MASLOW improves F1 by 5.8% on =1 tasks in cold-start (0.273→0.331) and 3.5% in complement (0.533→0.568). The cold-start gain is particularly significant, representing a 21% relative improvement where the system must generate useful training data from specifications alone. Decomposing this gain, context enrichment (SA_raw→SA) contributes the larger share (+5.2% in cold-start, +1.3% in complement), while multi-agent decomposition (SA→MASLOW) adds a further +0.6% and +2.2% respectively, demonstrating that both components contribute and that specialized agents outperform a single prompt even with the same enriched context; the decomposition gain is modest relative to enrichment, and we weigh it against its added cost in Section 6.

Naive feedback degrades quality. MASLOW+FB consistently underperforms MASLOW: −2.2% F1 on =1 tasks in cold-start, −1.9% in complement. The effect is more pronounced in cold-start, where no human labels provide compensating signal.

Data distribution analysis. To understand these performance gaps, we analyze four properties of the generated data (Section 4.5). All pipelines produce comparably diverse outputs, so diversity alone does not explain the differences. Instead, distributional alignment with human data better predicts downstream performance: MASLOW produces data closest to human distributions (Table 4). The feedback loop maintains diversity but disrupts label balance (positive ratio range 0.341–0.719 vs 0.430–0.515 without feedback; Table 5), explaining its consistent degradation. Paired Wilcoxon signed-rank tests [20] on per-class F1 (n=1,925) confirm all pairwise differences are statistically significant (Appendix G). All results above use zero-shot generation; few-shot generation improves cold-start substantially (+8.1% F1 on =1 tasks) but shows smaller or negative effects in complement (Appendix F).

4.3. Budget Allocation

The complement setting adds synthetic data on top of human labels, so gains could partly reflect additional training volume rather than data quality. To control for this, we fix total training size at 100 samples and vary the human-to-synthetic ratio (Table 2). Mixing is cost-effective: 20h+80s achieves F1=0.521 at $6.94/task (78% cost reduction versus $32/task human-only), exceeding the human-only baseline (F1=0.471, Table 1). Human labels remain more valuable per-sample, but the cost gap makes synthetic substitution attractive at scale.

Table 2
Fixed total training size of 100 samples per task (MASLOW, all 1,925 tasks). Cost estimated at $0.0067/synthetic sample and $0.32/human label (from Table 7).

Config	P	R	F1	Est. cost/task
100h + 0s	**0.699**	**0.599**	**0.645**	$32.00
80h + 20s	0.684	0.582	0.629	$25.73
60h + 40s	0.672	0.560	0.611	$19.47
40h + 60s	0.666	0.531	0.590	$13.20
20h + 80s	0.659	0.431	0.521	$6.94
0h + 100s	0.604	0.141	0.228	$0.67

4.4. Ablation Study

To isolate each agent's contribution without human-label confounding, we evaluate ablations in the cold-start setting (0h+200s) by removing one agent at a time. The Generator is not ablated because it is the core component; the other agents support it. Table 3 reports F1 drops relative to each full pipeline for both MASLOW and MASLOW+FB.

Table 3
Ablation: ΔF1 (%) vs full pipeline, cold-start (0h+200s). MASLOW+FB adds iterative feedback. MASLOW+FB w/o Validator $\equiv$ MASLOW w/o Validator, since removing it eliminates both filtering and feedback.

	MASLOW			MASLOW+FB		
	=1	<5	All	=1	<5	All
Full pipeline (F1)	0.331	0.334	0.246	0.309	0.323	0.223
w/o Extractor	−5.8	−4.6	−4.9	−3.6	−4.5	−3.3
w/o Analyzer	−2.7	−3.1	−3.0	−0.4	−0.5	+1.7
w/o Validator	−2.5	−1.1	−2.0	−0.3	0.0	+0.3

Extractor is the most critical component. Removing the Extractor so that downstream agents receive the original sparse task description instead of enriched context causes the largest drops for both pipelines (−5.8% and −3.6% F1 on =1 tasks), confirming that upstream context enrichment is most impactful in zero-shot generation. This large gap reflects the sequential architecture: degraded specifications propagate through all downstream agents, compounding errors at each stage.

Feedback loop undermines Analyzer and Validator contributions. In MASLOW without feedback, all three components contribute meaningfully (Extractor > Analyzer > Validator). In MASLOW+FB, the Analyzer and Validator contributions nearly vanish or reverse: removing the Analyzer *improves* F1 by 1.7% on all tasks, and removing the Validator has negligible effect. The feedback loop does not reduce diversity but causes label imbalance: MASLOW+FB positive ratios range 0.341–0.719 vs 0.430–0.515 without feedback. In MASLOW+FB, the Analyzer's guidelines and the Validator's iterative rejections create conflicting pressures that negate both components' contributions, particularly in cold-start where no human labels provide compensating signal.

4.5. Diversity and Distributional Analysis

We analyze within-dataset diversity, distributional overlap with human data, and label balance across pipelines (Tables 4 and 5). We embed each product's concatenated text fields using all-MiniLM-L6-v2 [21] and compute pairwise cosine distances. All synthetic pipelines use 200 samples per task; human uses 20 (matching the training configuration).

Diversity is negatively correlated with performance. Single-agent variants produce the most diverse outputs (0.504–0.529) but the lowest similarity to human data (0.345–0.369). MASLOW has the lowest diversity (0.451) but the highest similarity (0.388). This counter-intuitive finding suggests that distributional alignment with human data, rather than raw diversity, drives downstream performance. Higher diversity in SA variants likely reflects less constrained generation that explores regions of the product space irrelevant to the actual task, producing syntactically varied but semantically misaligned examples.

Table 4
Diversity, distributional alignment, and synth-only performance. Diversity: mean pairwise cosine distance (higher = more diverse). Synth F1: synth-only (0h+200s) micro F1 on all tasks. Similarity: 1 − mean cosine distance to human data (higher = closer). Brand recall: fraction of human brands recovered.

Pipeline	Within-dataset diversity		Alignment with human data	
	Diversity	Synth F1	Similarity	Brand recall
SA_raw	0.529	0.198	0.345	0.303
SA	0.504	0.218	0.369	0.323
MASLOW	0.451	0.246	0.388	0.326
MASLOW+FB	0.457	0.223	0.386	0.326
MASLOW-fs	0.491	0.412	0.408	0.498
MASLOW+FB-fs	0.498	0.407	0.406	0.479

Table 5
Per-task positive ratio across pipelines. All pipelines target 50/50 positive/negative split (positive ratio = 0.50).

Pipeline	Mean	Median	Min	Max
SA_raw	0.504	0.504	0.487	0.524
SA	0.504	0.505	0.485	0.524
MASLOW	0.491	0.493	0.430	0.515
MASLOW+FB	0.508	0.502	0.341	0.719

Few-shot generation improves distributional alignment. Providing seed examples dramatically increases overlap with human data: brand recall rises from 33% to 50% and synth-only F1 improves from 0.246 to 0.412. This confirms that seed examples steer generation toward the actual product distribution, and explains why few-shot generation produces large gains in the synthetic-only setting.

Feedback loop disrupts label balance without improving diversity. The feedback loop has minimal impact on diversity (MASLOW: 0.451 vs MASLOW+FB: 0.457) but causes substantially higher variance in per-task positive ratio: MASLOW ranges 0.430–0.515, while MASLOW+FB ranges 0.341–0.719. The Validator disproportionately rejects one label type per task, and iterative re-generation amplifies this skew rather than correcting it. This label imbalance, not reduced diversity or degraded individual sample quality, is the mechanism behind MASLOW+FB's consistent underperformance.

5. Experiments: In-Context Learning

Section 4 evaluated MASLOW on recalled product detection using supervised learning with PECOS. We now test whether findings generalize across two dimensions simultaneously: a different task type (product classification with balanced classes rather than recall detection with extreme imbalance) and a different learning paradigm (few-shot in-context learning rather than supervised training). ICL uses labeled examples as inference-time demonstrations with no parameter updates [16]. Its retrieval-based example selection makes it particularly suited to synthetic augmentation: a diverse generated pool enables query-adaptive selection of the most relevant demonstrations for each query, a capability that fixed human annotations cannot provide (Appendix H).

5.1. Experimental Setup

We evaluate on 40 unseen product classification tasks with balanced classes and abundant existing annotations (average 2,000 labels per task). We restrict access to 5 human labels per task to simulate cold-start conditions. Both SA and MASLOW generate 200 synthetic examples from the task specification alone (zero-shot), differing only in the generation pipeline. We report accuracy, F1, precision, and recall averaged across all 40 tasks.

Configurations. Each task has a fixed budget of 5 human-labeled examples. In brief, we compare zero-shot, 5-shot (human vs. retrieved-synthetic demonstrations), and 10-shot (human and synthetic combined) settings, and additionally test task-aligned feedback (AFB). Configurations are grouped by in-context shot count:

- **Zero-shot**: no demonstrations.
- **5-shot**: *Human-only* (5h) uses seed examples as fixed demonstrations. *SA* and *MASLOW* (5s) each retrieve the top-5 most relevant examples from their 200-example pools. Tests whether synthetic demonstrations can substitute for human ones.
- **10-shot** (5h+5s): combine 5 fixed human demonstrations with 5 retrieved synthetic examples. For MASLOW, we additionally evaluate *MASLOW+AFB* (Aligned FeedBack), which uses ICL's training-free inference to identify synthetic examples producing ambiguous predictions and feeds these back to the Generator for targeted re-generation.

5.2. Results

Table 6
Few-shot ICL performance averaged across 40 unseen product classification tasks with 5 human-labeled examples available. SA = single-agent baseline. Best results in **bold**.

Configuration	Accuracy	F1	Precision	Recall
Zero-shot	0.841	0.819	0.871	0.809
5-shot				
Human-only (5h)	0.857	0.838	0.879	0.821
SA (5s)	0.851	0.850	0.869	0.830
MASLOW (5s)	0.863	0.855	0.865	0.830
10-shot (5h + 5s)				
SA	0.871	0.860	0.872	0.830
MASLOW	**0.883**	**0.881**	**0.875**	**0.885**
MASLOW+FB	0.881	0.881	0.876	0.882
MASLOW+AFB	**0.890**	**0.885**	**0.884**	**0.886**

Synthetic data complements and can substitute for human labels. MASLOW complement outperforms both human-only and synthetic-only, yielding a +4.3% F1 gain over human-only. This parallels the supervised complementarity (+11.1% F1), confirming synthetic data adds value regardless of learning paradigm. MASLOW synthetic-only also exceeds human-only (+0.6% accuracy), validating that zero-shot generation with retrieval-based selection can produce more informative demonstrations than fixed human examples.

Multi-agent generation adds a consistent gain. MASLOW outperforms the single-agent baseline by +1.2% in both synthetic-only and complement settings. SA synthetic-only fails to match human-only, while MASLOW synthetic-only surpasses it, suggesting the full multi-agent pipeline is what pushes synthetic-only quality past human demonstrations in this setting, though the margin over SA is modest.

Task-aligned feedback improves ICL. In contrast to MASLOW+FB, which degrades supervised performance (Section 4), MASLOW+AFB rejects samples producing ambiguous predictions and improves ICL accuracy from 88.3% to 89%. This suggests that feedback effectiveness depends on alignment with the downstream task: naive feedback (FB) optimizes for generator-internal quality criteria (label correctness, attribute realism), which may not correlate with downstream utility. Aligned feedback (AFB) directly measures downstream impact by running candidate examples through the classifier and rejecting those that produce UNKNOWN predictions, ensuring that retained examples provide clear classification signal. The contrast between FB's consistent degradation and AFB's improvement opens directions for active-learning-inspired generation, where downstream task signals guide which synthetic examples to keep or regenerate.

6. Generation Cost and Latency

A practical synthetic data system must justify its cost against human annotation while maintaining acceptable latency. Table 7 reports generation cost and latency across pipeline configurations, estimated using public LLM API pricing with prompt caching enabled.

Table 7
Generation cost and latency (zero-shot, ∼400 samples/task, 1,925 tasks). Costs use public LLM API pricing with prompt caching. Per-task latency is sequential time for one task; wall-clock is parallel execution across concurrent tasks via the LLM API. Fail % = fraction of tasks that did not produce usable output.

	Cost				Latency		
Pipeline	**Est./task**	**Est. all 1,925**	**× cheaper**	**Fail %**	**Per task**	**All (parallel)**	**× faster**
SA_raw	$1.55	$2,984	83×	6.2%	18.1 min	0.8 hrs	40×
SA	$1.61	$3,099	80×	1.2%	19.7 min	1.0 hrs	37×
MASLOW	$2.75	$5,294	47×	0.7%	23.3 min	1.1 hrs	31×
MASLOW+FB	$3.69	$7,103	35×	1.3%	26.0 min	1.3 hrs	28×
Human label	$128	$246,400	1×	—	10–14 hrs	11 person-yrs	1×

Cost-performance tradeoff. MASLOW costs $2.75 per task, 71% more than the single-agent baseline ($1.61) but still 47× cheaper than human annotation ($128/task). The incremental cost of $1.14/task buys a +2.2% F1 gain on scarce tasks over SA. Context enrichment (SA_raw→SA) adds only $0.06/task for +2.0% F1, making it the highest-ROI component. MASLOW+FB costs $3.69/task (+34% over MASLOW) yet degrades quality, confirming the loss is mechanistic rather than a cost-quality tradeoff: the additional LLM calls for iterative re-generation do not recover their cost in performance.

Latency and parallelism. Each task requires 23.3 minutes of sequential LLM processing for MASLOW (four agent calls plus multiple generation batches). However, because tasks are independent, they can be processed concurrently via the LLM API. With sufficient parallelism, all 1,925 tasks complete in approximately 1.1 hours wall-clock, versus an estimated 11 person-years for equivalent human annotation. This makes same-day synthetic data generation feasible even at scale: a new batch of tasks arriving in the morning can have synthetic training data ready by the afternoon.

Reliability. The multi-agent architecture also improves generation reliability. SA_raw fails to produce usable output for 6.2% of tasks, typically due to insufficient context causing the LLM to generate off-topic or malformed samples. Adding context enrichment (SA) reduces failures to 1.2%, and the full MASLOW pipeline achieves the lowest failure rate at 0.7%. This is operationally significant: at 1,925 tasks, SA_raw leaves 119 tasks without synthetic data, while MASLOW leaves only 13.

7. Conclusion

We presented MASLOW, a multi-agent system that generates synthetic training data from sparse task specifications for low-resource e-commerce classification. Across 1,925 product-matching tasks and 40 unseen product classification tasks, we find that upstream context enrichment has outsized impact (−5.8% F1 when the Extractor is removed, twice the drop of any other agent), that multi-agent decomposition outperforms monolithic baselines given the same enriched input, and that synthetic data enables training across the full data-availability spectrum, from cold-start to complement to cost-effective budget allocation. Distributional alignment with human data, rather than raw diversity, better predicts downstream performance. Preliminary ICL experiments suggest that task-aligned feedback (MASLOW+AFB), which rejects samples causing ambiguous downstream predictions, can improve generation quality, opening directions for active-learning-inspired generation where downstream task signals guide data synthesis.

These findings offer practical guidance for teams building agentic synthetic data pipelines for e-commerce: (1) invest in input quality before optimizing generation architecture, as specification enrichment provides the highest return on both cost and performance; (2) prefer simple filtering over iterative refinement until better feedback mechanisms are developed, since naive feedback loops can introduce label imbalance that degrades downstream classifiers; and (3) when seed examples are unavailable, zero-shot generation with multi-agent decomposition remains viable for cold-start scenarios, while few-shot generation should be preferred whenever any labeled examples can be obtained. A key limitation is that synthetic data quality depends directly on specification quality: tasks with sparse or ambiguous specifications produce weaker synthetic data and correspondingly weaker classifiers, making

specification quality a prerequisite rather than a nice-to-have. Our evaluation also has limitations. The largest dataset cannot be released, limiting external reproducibility; we mitigate this by detailing the pipeline, prompts (Appendix C), and evaluation protocol. All four agents use the same underlying LLM, so any systematic bias of that model may propagate across stages rather than being corrected; heterogeneous-model pipelines remain an open question. Finally, our baselines isolate context enrichment and multi-agent decomposition but do not include a fully cost-matched multi-step baseline, so the marginal value of decomposition relative to its cost should be read as indicative rather than definitive. While our evaluation generates binary-labeled synthetic data (match/not-match per task), the Extractor-Analyzer-Generator-Validator pipeline applies to any setting where task specifications are underspecified and labeled data is scarce. Extending to synthetic data generation for multi-class classification tasks is a natural direction for future work.

References

[1] A. A. Barr, J. Quan, E. Guo, E. Sezgin, Large language models generating synthetic clinical datasets: a feasibility and comparative analysis with real-world perioperative data, Frontiers in Artificial Intelligence 8 (2025) 1533508.

[2] J. Gao, R. Pi, Y. Lin, H. Xu, J. Ye, Z. Wu, W. Zhang, X. Liang, Z. Li, L. Kong, Self-guided noise-free data generation for efficient zero-shot learning, arXiv preprint arXiv:2205.12679 (2022).

[3] H. Riaz, S. S. Bhabesh, V. Arannil, M. Ballesteros, G. Horwood, Metasynth: Meta-prompting-driven agentic scaffolds for diverse synthetic data generation, in: Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 18770–18803.

[4] M. K. Rad, A. Purpura, H. Kumar, E. Chen, M. S. Sorower, Graid: Synthetic data generation with geometric constraints and multi-agentic reflection for harmful content detection, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 30047–30065.

[5] H. Gupta, K. Scaria, U. Anantheswaran, S. Verma, M. Parmar, S. A. Sawant, C. Baral, S. Mishra, Targen: Targeted data generation with large language models, arXiv preprint arXiv:2310.17876 (2023).

[6] Z. Li, H. Zhu, Z. Lu, M. Yin, Synthetic data generation with large language models for text classification: Potential and limitations, arXiv preprint arXiv:2310.07849 (2023).

[7] M. Goyal, Q. H. Mahmoud, A systematic review of synthetic data generation techniques using generative ai, Electronics 13 (2024) 3509.

[8] Z. Qin, Q. Dong, X. Zhang, L. Dong, X. Huang, Z. Yang, M. Khademi, D. Zhang, H. H. Awadalla, Y. R. Fung, et al., Scaling laws of synthetic data for language models, arXiv preprint arXiv:2503.19551 (2025).

[9] Y. Meng, J. Huang, Y. Zhang, J. Han, Generating training data with language models: Towards zero-shot language understanding, in: Advances in Neural Information Processing Systems, volume 35, 2022, pp. 462–477.

[10] J.-E. Huang, I. Fang, T. Huang, C.-Y. Wang, J.-C. Chen, et al., Gen-n-val: Agentic image data generation and validation, arXiv preprint arXiv:2506.04676 (2025).

[11] J. D. Hastings, S. Weitl-Harms, J. Doty, Z. J. Myers, W. Thompson, Utilizing large language models to synthesize product desirability datasets, in: 2024 IEEE International Conference on Big Data (BigData), IEEE, 2024, pp. 5352–5360.

[12] A. Tavanaei, K. K. Koo, H. Ceker, S. Jiang, Q. Li, J. Han, K. Bouyarmane, Structured object language modeling (solm): Native structured objects generation conforming to complex schemas with self-supervised denoising, arXiv preprint arXiv:2411.19301 (2024).

[13] H. Tari, N. Sereiva, R. Kaushal, T. Bertaglia, A. Iamnitchi, Towards high-fidelity synthetic multi-platform social media datasets via large language models, arXiv preprint arXiv:2505.02858 (2025).

[14] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, in: International Conference on Learning Representations, 2021, pp. 1–12.

[15] M. Josifoski, M. Sakota, M. Peyrard, R. West, Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction, arXiv preprint arXiv:2303.04132 (2023).

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[17] J. Liu, D. Shen, Y. Zhang, W. B. Dolan, L. Carin, W. Chen, What makes good in-context examples for gpt-3?, in: Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures, 2022, pp. 100–114.

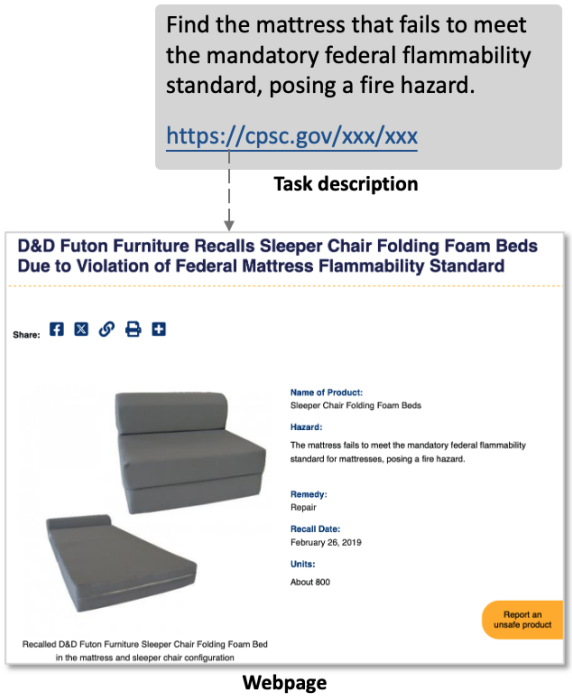
[18] H.-F. Yu, K. Zhong, J. Zhang, W.-C. Chang, I. S. Dhillon, Pecos: Prediction for enormous and correlated output spaces, 2022. URL: https://arxiv.org/abs/2010.05878. arXiv:2010.05878.

[19] Anthropic, Claude opus 4.6, https://docs.anthropic.com/en/docs/about-claude/models, 2025. Claude Opus 4.6.

[20] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945) 80–83.

[21] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, pp. 3982–3992.

A. Task Specification Enrichment Example



(a) Sparse Task Description

[Task description] Find the mattress that fails to meet the mandatory federal flammability standard, posing a fire hazard.

[URL Data] D&D Futon Furniture Recalls Sleeper Chair Folding Foam Beds Due to Violation of Federal Mattress Flammability Standard

Name of Product: Sleeper Chair Folding Foam Beds

Hazard: The mattress fails to meet the mandatory federal flammability standard for mattresses, posing a fire hazard.

Remedy: Repair

Recall Date: February 26, 2019

Units: About 800

(b) Enriched Task Description

Figure 4: Task specification enrichment example. (a) Sparse input containing only a brief description and URL. (b) Enriched specification after the Extractor fetches and parses the URL content into structured fields. The enriched specification still requires the Analyzer to determine discriminative attributes and decision boundaries.

B. Positive, Easy Negative, and Hard Negative Examples


```
{
  "item_name": "D&D Futon Furniture
    Chair Folding Foam Bed, Foam
    Density 1.8 lbs, Folded Sofa
    Mattress, 6 x 24 x 70, Gray",
  "bullet_point": "High quality density
    foam 1.8 pounds to increase the
    durability and resilience of the
    foam life. Convertible sleeper
    chair folds flat into a comfortable
    mattress. Perfect for guests,
    dorms, RVs, and small spaces.",
  "generic_keyword": "Sleeper Chair
    Folding Foam Beds, Flip Chairs,
    Sofa Beds, Flip Beds, Foldable
    Beds, Folding Mattresses, Sofa
    Couches, Portable Folded
    Mattresses, Floor Beds, RV
    Mattresses, Lounger Beds",
  "subject_keyword": "6 inch foam",
  "brand": "D&D Futon Furniture",
  "manufacturer": "D&D Futon Furniture",
  "label": "true"
}
```

(a) Positive

```
{
  "item_name": "Baby Einstein
    Musical Motion Activity Jumper
    with Sounds and Lights for
    Infants",
  "bullet_point": "Interactive
    baby activity center with
    music and lights |
    Developmental toys for babies
    6-12 months | Adjustable
    height settings | 360-degree
    rotating seat | Multiple toy
    attachments | Helps develop
    motor skills",
  "keyword": "baby activity center
    jumper musical developmental
    toys",
  "brand": "Baby Einstein",
  "manufacturer": "Kids II Inc",
  "label": "false"
}
```

(b) Easy Negative

```
{
  "item_name": "Milliard Tri-Fold
    Foam Folding Mattress and
    Sofa Bed for Guests - Queen
    78x58x4 Inches",
  "bullet_point": "Premium
    CertiPUR-US certified foam
    provides comfort and support.
    Tri-fold design allows easy
    storage. Removable soft
    microfiber cover is machine
    washable. Perfect for guests,
    camping, or RV use. Different
    brand and dimensions.",
  "keyword": "Tri-fold Mattress
    Folding Bed Guest Mattress
    Foam Bed Portable Mattress
    Sofa Bed Convertible Furniture
    ",
  "brand": "Milliard",
  "manufacturer": "Milliard",
  "label": "false"
}
```

(c) Hard Negative

Figure 5: Illustration of three sample types for a product-matching task. (a) Positive case matching all criteria. (b) Easy negative differing in multiple attributes. (c) Hard negative resembling the positive case but differing only in the discriminative attribute (brand), forcing classifiers to learn precise decision boundaries.

C. Sample Prompts

All prompts below are simplified for clarity. Implementation details (catalog identifier format, JSON schema, batch instructions) are omitted.

C.1. Extractor

```
[System]
You are a text parsing agent designed to extract information from
natural language text as structured output. A task is defined by
the task specification. Focus on accuracy and completeness.

[User]
Parse the following task specification to extract all URLs.
{{ TASK_SPECIFICATION }}

[Output]
{ "urls": ["https://...", ...] }
```

C.2. Generator

```
[System]
You are an expert product analyst. Generate realistic, diverse
synthetic products for training ML models to identify matching
products. Focus on accuracy, diversity, and creating genuinely
challenging negative examples.

[User]
Generate {TOTAL_SAMPLES} synthetic products: {NUM_POSITIVES}
True Positives + {NUM_NEGATIVES} Hard Negatives.
<task_specification> {TASK_SPECIFICATION} </task_specification>
<generation_guidelines> {GENERATION_GUIDELINES} </generation_guidelines>
<few_shot_examples> {FEW_SHOT_EXAMPLES} </few_shot_examples>

[Quality Requirements]
- Realistic e-commerce listing style (natural titles, detailed features)
- Diverse brands, manufacturers, and product variations
- Hard negatives must differ in only one or few critical attributes
- Include edge cases and borderline examples
- Vary terminology and phrasing across samples

[Feedback -- MASLOW+FB only]
If validator feedback is provided, analyze the patterns in
flagged issues and adjust generation strategy to avoid repeating
those errors. Generate entirely new samples.
```

C.3. Validator

```
[System]
You are an expert product analyst. Validate synthetic product
data to ensure it correctly represents products that match or
don't match the task specification.
```

```
[User]
Validate the following data against the task specification.
<task_specification> {TASK_SPECIFICATION} </task_specification>
<generation_guidelines> {GENERATION_GUIDELINES} </generation_guidelines>
<synthetic_data> {SYNTHETIC_DATA} </synthetic_data>

[Validation Criteria]
For each product, check:
1. Label accuracy: Does true/false match the specification?
2. Product realism: Does it sound like a real listing?
3. Feature consistency: Do attributes align with the label?
4. Category consistency: Does the product type match expectations?
IMPORTANT: Only report products with issues.

[Output]
{ "issues": [
    { "id": "...",
      "issue_type": "incorrect_label|unrealistic_product|...",
      "reasoning": "Detailed explanation and how to fix it" }
] }
```

C.4. Single-Agent Baseline

```
[System]
You are an expert product analyst. Complete three phases in sequence:
1. ANALYZE: Identify key characteristics for matching vs non-matching.
2. GENERATE: Generate realistic, diverse synthetic products.
3. VALIDATE: Check labels and remove problematic samples.

[User]
<task_specification> {TASK_SPECIFICATION} </task_specification>
<few_shot_examples> {FEW_SHOT_EXAMPLES} </few_shot_examples>

PHASE 1 - ANALYZE: Generate TRUE POSITIVE and HARD NEGATIVE
categories with descriptions, product types, and reasoning.

PHASE 2 - GENERATE: Using your analysis, generate {TOTAL_SAMPLES}
synthetic products ({NUM_POSITIVES} positives + {NUM_NEGATIVES}
hard negatives) with realistic e-commerce listing style.

PHASE 3 - VALIDATE: Check label accuracy, realism, and diversity.
Remove any samples with issues and replace them.

[Output]
<analysis> [All reasoning, categories, and validation notes]
</analysis>
<output> [Final validated JSON array only] </output>
```

D. Sample Count Sensitivity

Table 8 reports MASLOW F1 as a function of synthetic sample count in the synthetic-only setting.

Performance improves steeply from 50 to 150 samples, then plateaus: 200 to 250 samples offers marginal gains (+0.9% F1 on =1) at 25% higher generation cost. Beyond 250, F1 on scarce classes declines

Table 8
MASLOW F1 vs number of synthetic samples (synth-only 0h+Ns, zero-shot generation). Δ = change vs 50 samples.

Samples	F1 (=1)	Δ	F1 (<5)	Δ	F1 (all)	Δ
50	0.275	–	0.274	–	0.199	–
100	0.303	+2.8	0.309	+3.5	0.228	+2.9
150	0.327	+5.2	0.326	+5.2	0.240	+4.1
200	0.331	+5.6	0.334	+6.0	0.246	+4.7
250	0.340	+6.5	0.339	+6.5	0.250	+5.1
300	0.335	+6.0	0.336	+6.2	0.250	+5.1
350	0.341	+6.6	0.335	+6.1	0.253	+5.4
400	0.338	+6.3	0.335	+6.1	0.255	+5.6

slightly while the aggregate remains flat. We use 200 throughout as a practical cost-performance balance.

E. MRR Analysis: Single-Positive Tasks

For the 786 tasks with exactly one positive test sample, Mean Reciprocal Rank (MRR) measures how quickly the system surfaces the target product in its ranked predictions. Table 9 reports MRR and Recall@k across pipelines.

Table 9
MRR and Recall@k on 786 single-positive tasks. MRR = mean of 1/rank of the positive item. (so) = synth-only (0h+200s); others are mixed (20h+200s).

Pipeline	MRR	R@1	R@3	R@5
Human-20h	0.317	0.307	0.327	0.327
MASLOW	**0.465**	**0.447**	**0.485**	**0.485**
MASLOW+FB	0.455	0.439	0.471	0.471
SA	0.443	0.421	0.464	0.464
SA_raw	0.428	0.411	0.445	0.447
MASLOW (so)	0.220	0.209	0.230	0.230
MASLOW+FB (so)	0.203	0.195	0.213	0.213
SA (so)	0.212	0.199	0.227	0.227
SA_raw (so)	0.174	0.164	0.183	0.183

MASLOW improves MRR by +0.148 over human-only (0.465 vs 0.317), surfacing the target product at rank ∼2 on average. Recall@1 = 44.7% means nearly half of target products are ranked first. In cold-start (synth-only), MRR = 0.220 provides a useful initial ranking before human labels accumulate.

F. Zero-Shot vs Few-Shot Generation

Table 10 compares zero-shot generation (ZS: no seed examples provided to the Generator) against few-shot generation (FS: up to 20 human labels as seed demonstrations) across both the synthetic-only (0h+200s) and mixed (20h+200s) training settings.

In the synthetic-only setting, few-shot generation dramatically improves quality: providing seed examples improves F1 by 8–17%, confirming that seed availability is a more important lever than iterative refinement (<2.3% F1 effect). This aligns with the distributional analysis in Section 4.5: seed examples ground generation in the actual product distribution, providing stronger signal than any post-hoc filtering.

In the mixed setting, the effect reverses for the most scarce tasks: ZS outperforms FS for MASLOW on =1 (−2.9%) and <5 (−1.7%) strata, though FS improves on all tasks (+1.0%). This suggests that when human labels are available for training, zero-shot generation produces more complementary synthetic

Table 10
Zero-shot (ZS) vs few-shot (FS) generation across training settings. FS provides up to 20 human labels as seed demonstrations. MASLOW+FB adds an iterative feedback loop.

Training	Gen.	Pipeline	=1 label (786)			<5 labels (1,277)			All (1,925)		
			P	R	F1	P	R	F1	P	R	F1
Synth-only	ZS	MASLOW	0.590	0.230	0.331	0.581	0.234	0.334	0.608	0.154	0.246
	FS	MASLOW	0.598	0.314	**0.412**	0.590	0.432	**0.499**	0.574	0.313	**0.405**
	ZS	MASLOW+FB	0.566	0.212	0.309	0.556	0.227	0.323	0.606	0.136	0.223
	FS	MASLOW+FB	0.588	0.312	**0.407**	0.585	0.436	**0.499**	0.545	0.299	**0.386**
Complement	ZS	MASLOW	0.686	0.485	**0.568**	0.710	0.619	**0.661**	0.663	0.440	0.529
	FS	MASLOW	0.653	0.459	0.539	0.668	0.623	0.644	0.633	0.469	**0.539**
	ZS	MASLOW+FB	0.660	0.471	**0.549**	0.694	0.612	**0.650**	0.667	0.435	0.527
	FS	MASLOW+FB	0.661	0.457	0.540	0.677	0.621	0.648	0.620	0.455	0.525

data, likely because FS-generated examples overlap more with the human labels already in the training set, reducing the diversity benefit. The practical implication is clear: when deploying in true cold-start (no labels available), use few-shot generation if any seed examples can be obtained; when augmenting existing human labels, zero-shot generation provides better complementarity.

G. Statistical Significance Tests

Table 11 reports paired Wilcoxon signed-rank tests [20] on per-task F1 across 1,925 tasks in the mixed training setting (20h+200s).

Table 11
Pairwise statistical comparisons (mixed setting, 20h+200s, per-task F1). Δ = difference in mean F1. Cohen's d: $|d| < 0.2$ negligible, 0.2–0.5 small.

Comparison	A mean	B mean	Δ (B–A)	p-value	Cohen's d
SA_raw $\rightarrow$ SA (enrichment)	0.525	0.536	+0.011	2.04e-02*	+0.057
SA $\rightarrow$ MASLOW (decomposition)	0.536	0.552	+0.016	1.83e-09***	+0.086
MASLOW $\rightarrow$ MASLOW+FB (feedback)	0.552	0.539	-0.012	2.01e-06***	-0.071
SA_raw $\rightarrow$ MASLOW (combined)	0.525	0.552	+0.027	1.21e-14***	+0.130
Human-20h $\rightarrow$ Human+MASLOW	0.471	0.552	+0.081	1.95e-38***	+0.304

All pairwise differences are statistically significant. The largest effect is the synthetic complement over human-only (Cohen's d=0.304, small effect), followed by the combined enrichment and decomposition effect (d=0.130).

H. In-Context Learning Input Example

Simplified prompt structure for product classification.

```
[System Prompt -- same for all inputs]
Task: Classify products against a task specification
to determine if they match a specified category.
Instructions:
1. Review specification components in tags:
   <task_description>: Matching category definition
   <query_expansion>: Additional matching criteria
   <keywords_of_interest>: Indicative keywords
   <keywords_not_of_interest>: Exclusion keywords
2. Evaluate predicates using Boolean logic (AND/OR).
3. Output: [Step-by-step analysis]
<classification_reason>[50-word summary]</classification_reason>
```

```
[MATCH/NOMATCH/UNKNOWN]

[Task-Specific Specification]
<task_description>
    Coffee beans and grounds: roasted coffee seeds and ground
    powder used to brew coffee.
</task_description>
<keywords_of_interest>
    coffee beans, coffee grounds, brewed coffee
</keywords_of_interest>
<keywords_not_of_interest>
    furniture, electronics, apparel, shoes
</keywords_not_of_interest>

[Query Product]
{
    'item_name': "morning person coffee maker t-shirt",
    'product_group': 'apparel',
    'product_type': 'shirt',
    'brand': 'finest prints'
}

[Few-Shot Example]
<exemplar_0>
    <exemplar_product>
        {
            'item_name': "keep calm love brewed coffee t-shirt",
            'product_group': 'apparel',
            'product_type': 'shirt'
        }
    </exemplar_product>
    <exemplar_answer>NOMATCH</exemplar_answer>
</exemplar_0>
```