Autoresearch for Marketplace Catalogs: From Legacy Forms to AI-Native Matching

Kartik Ravisankar
kravisankar@thumbtack.com
Thumbtack
San Francisco, CA, USA

Hojat Abdolanezhad
habdolanezhad@thumbtack.com
Thumbtack
San Francisco, CA, USA

Daniel Capo
dcapo@thumbtack.com
Thumbtack
San Francisco, CA, USA

Sang Su Lee
psulee@thumbtack.com
Thumbtack
San Francisco, CA, USA

Shishir Dash
shishirdash@thumbtack.com
Thumbtack
San Francisco, CA, USA

Vijay Anand Raghavan
vraghavan@thumbtack.com
Thumbtack
San Francisco, CA, USA

Abstract

Two-sided service marketplaces that match consumers to service providers are transitioning from deterministic request-form intake to an AI-native probabilistic matching system, enabled by recent advances in large language models (LLMs) that can infer user intent, contextual preferences, and latent constraints from natural language instructions. As marketplaces increasingly rely on inferred intent rather than fixed-form fields, these platforms must regenerate the provider-side preference taxonomy that underwrites matching, search, and pricing: a set of provider attributes and preferences interpretable to service providers while remaining useful signals for marketplace decision-making. We present an autoresearch loop that generates this taxonomy one occupation at a time. The system has been deployed in production at a major U.S. consumer services marketplace since April 2026 across 132 occupations. Instead of constructing a single global hierarchy, the loop treats each occupation as an independent generation problem and runs iterative propose-evaluate-keep refinement cycles. Each candidate tag set is scored using a recalibrated six-rubric LLM-as-judge framework, producing a composite score of up to 15. A 7-critic panel, each with distinct personas, contributes weighted penalties to produce an adjusted score, with no hard vetoes. A separate LLM-based parity-mapping stage maps legacy request-form Q&A pairs back to the generated taxonomy, producing both a coverage signal and a scalable interface for human quality assurance. The loop migrates legacy structured Q&A by first inferring the underlying provider attribute that each question was intended to measure, rather than performing literal question-to-tag translation. Unlike prior autonomous taxonomy-generation systems, our approach (i) takes the per-occupation preference-tag catalog (not the hierarchy of occupations) as the unit of optimization, (ii) generates each occupation's tags independently in parallel, (iii) evaluates with marketplace-grounded critics contributing weighted penalties (no hard vetoes) to an adjusted score, and (iv) treats legacy-Q&A-to-modern-tag migration as a distinct pipeline component rather than a downstream side effect. We report deployment results from a 14-day post-launch production cohort (1,840 enrolled pros, 9.3M filter evaluations) that surfaces a concrete catalog-hygiene gap that a global-build approach would have masked.

Keywords

LLM autoresearch, taxonomy generation, marketplace catalog, multi-critic evaluation, LLM-as-judge, deployed system

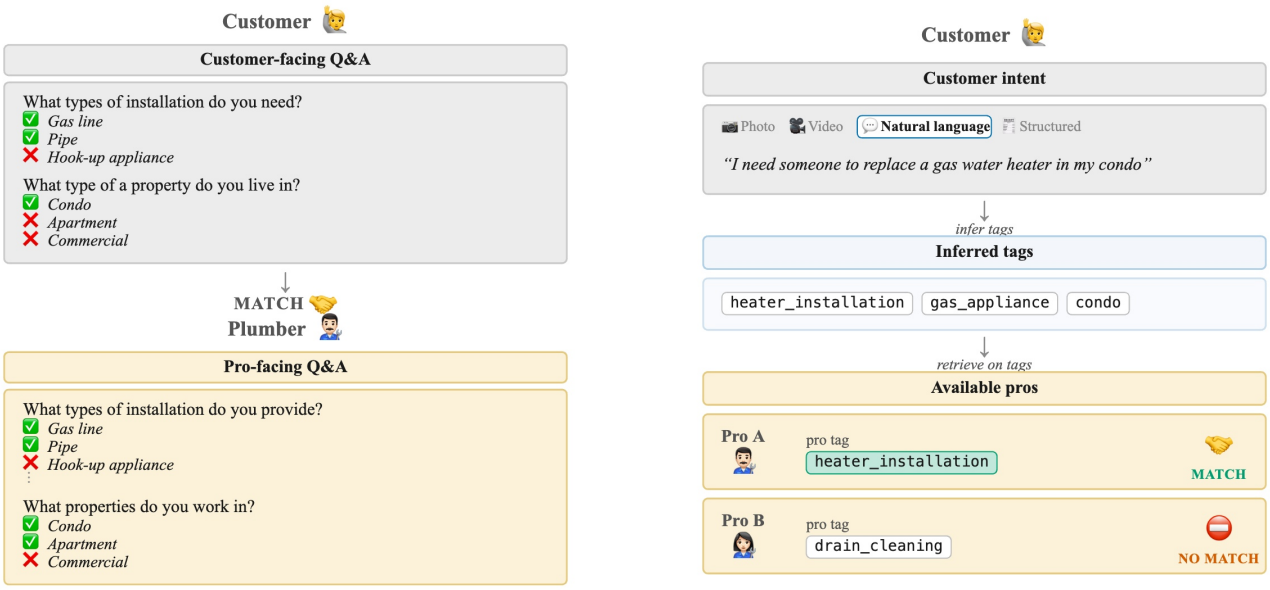
1 Introduction

A marketplace catalog provides the shared representation through which consumer requests, provider preferences, and job attributes are interpreted. Matching, search, and pricing all depend on this common language. However, in service marketplaces that have historically relied on structured question-and-answer (Q&A) forms, the catalog is often implicit, distributed across hundreds of category-specific schemas rather than represented as a unified structure. When the marketplace transitions to AI-native probabilistic matching (Figure 1), this implicit catalog becomes a liability. Matching models require structured signals, pricing models require job attributes that transfer across categories, and marketplace participants need a coherent interface for expressing preferences without navigating hundreds of independent schemas.

We frame this problem as a catalog *reconstruction*, not *creation*, since the catalog is already implicitly defined in the legacy system, albeit distributed, through Q&A pairs. The task is to infer the latent attributes encoded across Q&A pairs and reorganize them into an explicit representation suitable for AI-native marketplace interactions. Because the implicit legacy catalog is organized around distinct service domains, each with its own vocabulary, preference structure, and matching semantics, we decompose catalog reconstruction into a collection of smaller *autoresearch* problems [8] rather than a single global taxonomy-generation task. This design contrasts with recent hierarchical taxonomy-generation systems, which optimize coherent occupation hierarchies [9] or corpus-level research taxonomies [7]. Our goal is not to build a single hierarchy over all domains, but to reconstruct preference catalogs whose semantics remain meaningful within each service domain in a two-sided marketplace.

The key contributions of this paper are as follows:

(1) **Catalogs for marketplaces -** We study a catalog- reconstruction problem distinct from occupation-hierarchy construction [9], text-label taxonomy induction [12], research-corpus taxonomy adaptation [7], and product-schema modeling [5]. The generated unit is a provider-facing *preference tag*: a primitive that providers toggle with in production and that the downstream marketplace systems use.



(a) Legacy Q&A matching **(b) AI-native matching**

Figure 1: Evolution from legacy request-form matching to AI-native matching: Legacy matching (a) relies on an implicit catalog embedded in occupation-specific Q&A forms. AI-native matching (b) requires an explicit catalog that translates consumer intent and provider preferences into a shared representation for downstream marketplace systems.

(2) **Independent autoresearch loops -** We decompose reconstruction into independent autoresearch jobs, one per service occupation, each with its own *propose-evaluate-keep* loop. This avoids forcing all service domains into a single global hierarchy and limits semantic interference between occupations with superficially similar but operationally distinct attributes.
(3) **Marketplace-grounded automated evaluation -** We combine a recalibrated multi-rubric LLM-as-judge [14] with a seven-persona critic panel that contributes weighted penalties to the generated catalog. This aligns evaluation with marketplace objectives rather than generic measures of taxonomy quality.
(4) **Legacy Q&A parity mapping -** We introduce a parity-mapping stage that maps the reconstructed catalog back to legacy request-form Q&A pairs. This stage measures whether the new catalog covers the operational distinctions encoded in the legacy system and produces a human-reviewable artifact for validating migration from structured forms to provider-preference tags.

Success for this system means the marketplace can retire its legacy request forms without losing the preference structure they encoded. The most direct evidence is reconstruction fidelity. Across the reconstructed catalog, 73.3% of legacy answers map directly to a tag in the regenerated catalog; most of the remainder are intake-only answers (sizes, ranges, "flexible") that a preference schema should not carry, excluded as acceptable non-tags (20.8% of all answers). Only 6.0% of legacy answers are *regrettable* misses, real preferences left uncovered. Measured against the preferences that should map, pooled coverage is 92.5%, the per-occupation median is 100% (mean 97.8%). What the catalog drops is, by construction, intake detail; the screening preferences pros act on are almost entirely preserved. Together with production scale (132 occupations live, new ones onboarded on demand within hours) and mandatory human sign-off before deployment, this fidelity is what makes retiring the request forms defensible.

The propose-evaluate-keep loop builds on prior work in iterative LLM refinement, principle-guided critique, and autoresearch systems [2, 8, 10]; here, we operationalize it for production catalog reconstruction in a two-sided service marketplace. We report results from a production deployment spanning 132 service occupations as of April 2026, with on-demand onboarding of new occupations as the marketplace expands. The remainder of the paper describes the marketplace setting (§2), other related works (§3), the autoresearch system used for catalog reconstruction (§4), its deployment in production (§5), and the empirical findings (§6) that emerge from operating the system at scale.

2 Background

In this section, we describe the marketplace setting and the production context in which the system is deployed (§2.1), as well as the terminology needed to understand the catalog-reconstruction problem (§2.2).

2.1 Marketplace context

The deploying organization is a major U.S. online services marketplace that connects consumers with local service professionals (*pros*) across home, wellness, and event categories, including plumbing, house cleaning, photography, landscaping, and roughly 130 others. Pros specify targeting preferences that determine which consumer leads they are eligible to receive and purchase. The system described in this paper is part of a 2026 pro-side marketplace redesign that replaces rigid binary targeting defined through structured Q&A with probabilistic matching driven by a structured provider-side preference taxonomy [4].

2.2 Terminology

The legacy taxonomy is organized as a hierarchy of occupations, categories, and category-specific request-form Q&A schemas. The reconstructed catalog contains two tag classes.

(1) *Canonical tags* determine eligibility: a provider must possess the canonical tag associated with a category for the strict-match filter to consider that provider eligible for leads in that category;
(2) *Specialty preference tags* allow providers to refine which leads they wish to receive within categories for which they are already eligible

Section 6 reports two production findings that depend on this distinction: a catalog-hygiene gap in which deprecated canonical tags were still emitted by request-time enrichment, and a deployment-side seeding gap in which 40.66% of *filter evaluations* failed because the canonical tag associated with the consumer request was not present on the provider profile. A *filter evaluation* is a single matchmaker-side check, for one candidate pro and one incoming consumer request, to determine whether the pro's profile contains the request's required canonical tag.

3 Related Work

Our work sits at the intersection of automated taxonomy construction, agentic schema generation, autoresearch systems, prompt optimization, and LLM-based evaluation.

Hierarchical taxonomy construction. CLIMB [9] built occupation hierarchies using a global semantic clustering to distill core occupations, followed by a reflection-based multi-agent system to iteratively build a coherent hierarchy. Our work differs in both the unit of generation and the reconstruction objective: CLIMB generated occupations within a hierarchy, whereas we reconstruct provider-preference tags within an occupation. Furthermore, our catalogs are generated independently for each occupation and must support migration from legacy request-form Q&A systems through parity mapping. TnT-LLM [12] used LLMs to induce and iteratively refine label taxonomies from unstructured text, while TaxoAdapt [7] dynamically adapted multidimensional taxonomies to evolving scientific corpora through iterative hierarchical classification. Both systems inform the generation-and-refinement paradigm we adopt. However, their objective is taxonomy induction for corpus organization and classification, whereas ours is catalog reconstruction for marketplace decision-making.

Schema and catalog generation. AttributeForge [5] automated end-to-end product-schema modeling using a large collection of specialized LLM agents, together with automated evaluation and repair. It is the closest prior work to ours in terms of production-scale catalog reconstruction. The key distinction is the object being modeled: AttributeForge generated product attributes for e-commerce catalogs, whereas we reconstruct provider-preference catalogs for service marketplaces, where matching depends on both provider capabilities and provider screening preferences.

Autoresearch, self-refinement, and prompt optimization. Our "propose, evaluate, keep" loop draws on a broader family of iterative LLM optimization methods, including Self-Refine [10], Constitutional AI [2], multi-agent debate [3], GEPA [1], and prompt-optimization frameworks such as DSPy and MIPROv2 [11]. Like DSPy, we treat prompts as optimizable artifacts and evaluation as the search signal. However, rather than optimizing a single prompt against a fixed benchmark or held-out metric, our framework optimizes occupation-specific catalog-generation prompts whose outputs are structured marketplace catalogs.

LLM-based evaluation and abstraction. Our system draws on recent work in both LLM-based evaluation and abstraction. The six-rubric evaluation framework builds on LLM-as-judge methodologies [14] and critique-based evaluation [6], adapting them to catalog reconstruction through marketplace-specific criteria derived from production review. Several rubrics were introduced to capture failure modes that generic taxonomy metrics overlook, such as tags that are semantically coherent but insufficiently specific or interpretable for provider-facing use. We also draw inspiration from Step-Back Prompting [13]. Rather than translating legacy request-form questions directly into tags, we first infer the underlying job attribute each question was intended to capture and use that abstraction as input to catalog generation.

4 System Design

We formulate catalog reconstruction as an iterative propose-evaluate-keep problem (§4.1). Candidate catalogs are generated independently for each service domain, evaluated using a rubric-based LLM judge (§4.2), adjusted through a multi-persona critic panel (§4.4), and validated against legacy request-form semantics through parity mapping (§4.6).

4.1 Per-Occupation Autoresearch Loop

Catalog reconstruction is a cold-start problem: initially there is no per-occupation generation prompt, so every occupation $o \in O$ ($|O| = 132$ in the current production deployment) is seeded with the same baseline prompt $p^{(0)}$, i.e. $p_o^\star \leftarrow p^{(0)}$. We run Algorithm 1 (illustrated in Figure 2) independently per occupation.

At iteration 0, the generator $\mathcal{G}$ applies the baseline prompt to the occupation's legacy data D_o (RF Q&A schema for each category, % share of recently-active pros who have currently enrolled a (q, a) pair as a preference) to produce a candidate tag set T. The six-rubric LLM-as-judge scores the set, $s = \mathcal{E}(T) \in [0, 15]$, and the seven-persona critic panel adds weighted penalties π_k, giving the composite score $a = s - \sum_k w_k \pi_k(T)$. This becomes the initial best $(p_o^\star, T^\star, s^\star, a^\star)$. The loop runs a cross-family model stack: the

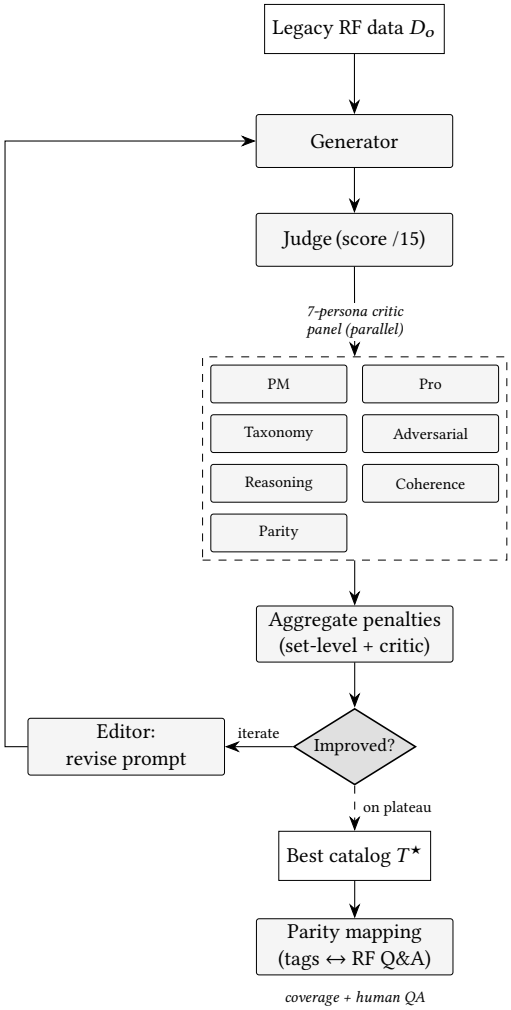


Figure 2: Per-occupation autoresearch loop (conceptual overview; see Algorithm 1 for the precise procedure). The occupation's legacy RF data D_o and current best prompt p_o feed a Generator that produces a candidate tag set. Each candidate is scored by the $\mathcal{E}$ six-rubric LLM-as-judge; the seven-persona critic panel then runs in parallel, and its feedback is aggregated. The gate decides whether the editor's single proposed prompt edit improves on the current best and is kept. On a plateau or budget exhaustion, the best catalog $T^\star$ is handed to the post-loop parity mapping, which links catalog tags to legacy RF Q&A (many-to-many) and produces the artifact humans review.

generator $\mathcal{G}$ and judge $\mathcal{E}$ on one family (GPT-5-4 / GPT-5-4-mini), the seven critics, editor, and parity mapper on another (Claude Sonnet 4.6), so the models that propose and score do not share biases with the models that critique and mutate (§4.2, §7).

Each subsequent iteration proposes and tests a single prompt edit. The editor $\mathcal{R}$ makes one targeted change to the current best prompt $p_o^\star$, focused on a single weak quality dimension: it ranks the six

Algorithm 1 Occupation-Local Autoresearch

Require: legacy data D_o with RF Q&A Q_o, shared seed prompt $p^{(0)}$, budget B, critic weights w
Ensure: best prompt $p_o^\star$, best catalog $T^\star$, parity relation $M^\star$, unmapped sets $U_Q^\star, U_T^\star$

$p_o^\star \leftarrow p^{(0)}$, $T^\star \leftarrow \mathcal{G}(p_o^\star, D_o)$, $a^\star \leftarrow \mathcal{E}(T^\star) - \rho(T^\star) - \sum_k w_k \pi_k(T^\star)$
for $t = 1$ to B **do**
  $p_o \leftarrow \mathcal{R}(p_o^\star)$, $T \leftarrow \mathcal{G}(p_o, D_o)$, $\hat{a} \leftarrow \mathcal{E}(T) - \rho(T)$ ▷ pre-critic score; ρ = set-level penalty (parity coverage + proliferation)
  if $\hat{a} > a^\star$ **then** ▷ $\hat{a} \geq a$, so a candidate with $\hat{a} \leq a^\star$ provably cannot win; skip critics
    $a \leftarrow \hat{a} - \sum_k w_k \pi_k(T)$
    if $a > a^\star$ **then**
      $(p_o^\star, T^\star, a^\star) \leftarrow (p_o, T, a)$
    end if
  end if
end for
$M^\star \leftarrow \mathcal{P}(T^\star, Q_o) \subseteq T^\star \times Q_o$ ▷ many-to-many tag ↔ Q&A relation
$U_Q^\star \leftarrow Q_o \setminus \pi_Q(M^\star)$ ▷ unmapped Q&A
$U_T^\star \leftarrow T^\star \setminus \pi_T(M^\star)$ ▷ tags with no covering Q&A
return $(p_o^\star, T^\star, M^\star, U_Q^\star, U_T^\star)$

$\mathcal{E}$ rubrics together with an injected parity-coverage dimension by mean score and selects the lowest-scoring dimension not targeted in the previous two iterations, avoiding fixation on any one rubric. The editor conditions on the weakest tag examples in that dimension, the score trends across iterations, and the log of previously accepted and rejected edits. The revised prompt is regenerated and rescored; as a per-iteration cost saver, the critic panel is invoked only when the candidate's $\mathcal{E}$ score is not already below the best ($s \geq s^\star$), since a lower-$\mathcal{E}$ candidate is very unlikely to overcome the best composite score. The edit is accepted, thus becoming the new best, *iff* its composite score improves, $a > a^\star$; otherwise the prompt reverts to $p_o^\star$. The loop runs for a fixed budget of B iterations (default 5). Appendix B traces one iteration end-to-end on the *Accounting* occupation.

On termination, the parity mapping $\mathcal{P}(T^\star, Q_o)$ links the tags of the best catalog $T^\star$ to the legacy RF Q&A Q_o as a many-to-many relation $M^\star \subseteq T^\star \times Q_o$, leaving residual sets $U_Q^\star$ (RF Q&A covered by no tag—typically intake-specific size/range/flexibility questions) and $U_T^\star$ (tags covered by no Q&A). We emit $M^\star$ as a JSON mapping and a Google Sheet for human review before deployment.

4.2 Six-Rubric LLM-as-Judge

The judge model $\mathcal{E}$ scores every tag ($t \in T$) on six dimensions:

- **Screening vs. Intake** (0–3): Does the tag describe a job type the pro might want *more or less of* (good), or is it a customer-intake detail the pro should not screen on (bad)? E.g. "Tax return preparation" (screening) vs. "Travel range: 15 miles" (intake).

- **Tag Legibility** (0–3): Would a working pro understand this tag from the tag text alone? E.g. "Appliance repair" (clear) vs. the bare noun "Appliances"
- **Preference Variance** (0–3): Do real pros split on this tag, or is everyone equally for-or-against (degenerate)? E.g. Accounting pros split on "QuickBooks" proficiency, but nearly all accountants serve the "healthcare industry" (degenerate).
- **Cross-Category Consistency** (0–2): When the same job-attribute appears in two categories within the occupation, do the tags match? E.g. the client-entity type tagged identically as "S-Corp" wherever it recurs across the occupation's categories, not "S-Corp" in one and "S-Corporation" in another.
- **Canonical Coverage** (0 or 2): Does every category in the occupation have a canonical tag covering it? E.g. "Payroll services" tag cleanly covers the *Payroll Services* category, whereas a tag backed by no category scores 0
- **Information Loss** (0–2): Are the tag's sources traceable to specific legacy Q&A answers, i.e. does it cite the legacy category and answers it consolidates? E.g. a tag citing legacy category: Accounting and the "QuickBooks" answer scores 2 vs. one with only a vague rationale (0).

$\mathcal{E}$ scores each tag independently on the six rubrics above, with all tags in T scored in parallel; the per-tag composite is the sum of the six rubric scores (on a /15 scale). The set-level score is the mean per-tag composite reduced by small set-level deductions for tag proliferation and category-coverage gaps, $s = \mathcal{E}(T) \in [0, 15]$; the verdict mix summarizes the per-tag outcomes. The judge model is GPT-5-4-mini at temperature zero; the generation model is GPT-5-4. The seven-persona critic panel, the editor agent, and the final parity-mapping stage all run on Claude Sonnet 4.6, a different model family from the generator and the judge. The choice is deliberate: a same-family stack would risk the loop reinforcing biases shared across mutation and evaluation, and the cross-family arrangement is the strongest readily available defense at production scale. $\mathcal{E}$ was recalibrated against product-manager (PM) review: bare-noun tags like "Appliances" originally scored a legibility of 3 but scored ≤ 1 after recalibration, since a working appliance-repair pro reads "Appliances" as either trivially true or meaningless, not informative.

4.3 Set-Level Penalty (ρ)

The $\mathcal{E}$ judge scores each tag in isolation, so a catalog of individually high-scoring tags can still be poor as a *set*. T can be bloated with near-duplicate siblings, or leave real pro preferences with no corresponding tag. The set-level penalty $\rho(T)$ corrects for this by deducting from the composite two failures the per-tag rubrics cannot see: *tag proliferation* (an excess of sibling tags under a single parent, which fragments the catalog and the pro-facing UI) and *coverage gaps* (legacy preferences in Q_o which pros both enrolled and deselected recently that no tag captures, including whole categories left untagged; §4.6). Subtracting $\rho(T)$ before the critic penalties (§4.4), $a = s - \rho(T)$, makes the loop optimize for a coherent, well-covered catalog rather than a collection of locally excellent but globally redundant or incomplete tags.

4.4 Seven-Persona Critic Panel

After $\mathcal{E}$ scoring, the full tag set is reviewed by seven critic-personas running in parallel, each returning a verdict that is reduced to a weighted contribution to the composite score. Most contributions are penalties; the Reasoning critic can instead award a small bonus for strong rationales. The exact penalty formulas, caps, and bonus thresholds are tabulated in Appendix A.[1] The critics are:

- **PM critic**: Would a product manager (PM) find this tag actionable? Surfaces design-stage issues (mutual-exclusivity, awkward dimensionality); penalized only when rejections exceed a small free allowance.
- **Pro critic** (penalty capped at 2.5): Would a working pro in this occupation find this useful or confusing? The cap prevents a single skeptical persona from dominating the verdict.
- **Taxonomy critic**: Are tags at consistent levels of abstraction, are unions broken into components, are subtypes grouped? Penalized as structural consistency falls below the threshold.
- **Adversarial critic**: Where could the tag set be gamed or misinterpreted to extract leads that the pro cannot deliver? Penalty scales with the assessed risk level and the number of critical failures.
- **Reasoning-Quality critic** (bonus-eligible): Does the generation explain *why* each tag exists? Strong rationales (score ≥ 8) earn a small bonus; weak ones incur a penalty.
- **Occupation-Coherence critic**: Does the tag set, taken as a whole, describe *this occupation* rather than a generic union of services?
- **Parity critic**: Do the generated tags map to a given occupation's top-30 most-deselected legacy answers (i.e., answers active pros have explicitly opted out of), ranked by distinct-pro count over a 90-day window? Its penalty is small by design (at most $\sim$ 1 point); the quantitative coverage enforcement lives in the set-level penalty ρ (§4.3). When an occupation has no deselection signal, the critic returns a neutral score and contributes nothing.

4.5 Editor Agent

The editor $\mathcal{R}$ is the loop's mutation operator: at each iteration, it proposes exactly one targeted edit to the current best prompt $p_o^\star$ and returns the revised prompt together with a one-line description of the change. A key design point is that the editor is driven by the judge $\mathcal{E}$, not the critics: the critic penalties gate which prompts survive (through the composite a), whereas the editor decides *what to change* from the $\mathcal{E}$ signal and the coverage signal alone.

Target-dimension selection. The editor ranks two complementary kinds of signals and targets the weakest. The first is *per-tag quality*: the six $\mathcal{E}$ rubrics, each averaged across the tag set to yield one mean per rubric. The second is *set-level coverage*: a single parity-coverage

[1] An earlier design used a hard veto from the Pro Critic. In early production runs, the veto dominated keep/discard decisions, and the loop stalled on occupations where one persona consistently rejected proposals that other critics rated favorably. The current design replaces the veto with a weighted penalty capped at 2.5 on the composite-of-15 scale; low enough that a strong proposal can still be kept over Pro objections, high enough that consistent Pro disapproval pulls the adjusted score below the discard threshold. The cap and rate were set empirically; the panel module records the change as "softened to weighted penalty, capped at 2.5."

score for the catalog as a whole. Coverage cannot be expressed as a seventh rubric, since $\mathcal{E}$ scores the tags that exist, whereas coverage is a property of the legacy preferences that have *no* tag, so there is nothing per-tag to score. It is therefore injected as a synthetic dimension, giving catalog incompleteness a seat in the ranking alongside tag-quality weaknesses. The coverage dimension is a *score* (higher is better), $2\min(r_+, r_-) \in [0, 2]$, where $r_+, r_- \in [0, 1]$ are the fractions of enrolled and deselected preferences the catalog covers; the 0–2 scaling puts it on the same footing as the rubric means. This ranking score is used *only* to select the target dimension and is distinct from the set-level penalty ρ (§4.3), which turns the same coverage signal into points subtracted from the composite. The editor then targets the lowest-scoring dimension that has not already been addressed in the previous N iterations (default N = 2), a constraint that prevents fixation on a single persistent weakness and forces progress across both quality and coverage rather than over-optimizing one axis.

Conditioning. For the chosen dimension, the editor is given the tags scoring weakest on it ($\mathcal{E}$ rationales with a score ≤ 1, or the specific unmapped preferences when parity-coverage is targeted), the dimension's score trend across iterations (improving / stable / worsening), and a log of prior edits annotated with their kept/discarded outcomes. It is instructed to propose exactly one change, to try a fundamentally different approach when a dimension keeps worsening despite edits targeting it, and never to repeat a previously discarded change. The revised prompt $p_o^{(t+1)}$ is then regenerated and rescored (§4.1), and kept only if its composite score improves.

4.6 Parity mapping

Once the loop terminates with the best catalog $T^\star$, a final LLM pass migrates the occupation's legacy RF Q&A Q_o onto the new tags, producing the many-to-many relation $M^\star \subseteq T^\star \times Q_o$ returned by Algorithm 1. The autoresearch design ensures that the generated catalog covers a good fraction of legacy taxonomies (explicitly in §4.3, parity critic in §4.4, and set level coverage in §4.5). $M^\star$, on the other hand, is *explicit and auditable*: it records exactly which legacy preferences map to which tags and which remain uncovered, producing the human-reviewable artifact used to sign off the catalog before deployment. For each legacy answer, the mapper finds *all* tags that cover the same real-world concept; a single answer may map to several tags (for example, a specific tag and the broader tag above it), which is why $M^\star$ is a many-to-many relation rather than a function. Aggregated across the 132-occupation production catalog, $M^\star$ is also what yields the reconstruction-fidelity numbers quoted in §1: 73.3% of legacy answers map directly to a tag in the regenerated catalog; 20.8% are intake-only answers (sizes, ranges, "flexible") excluded as acceptable non-tags; the remaining 6.0% are regrettable misses, real preferences left uncovered. Measured against the answers that should map, pooled coverage is 92.5%, with a per-occupation median of 100% (mean 97.8%).

5 Deployment

The reconstructed catalog has been running in production since April 2026. This section describes how catalogs were generated at scale across the occupation set and how each catalog was staged for human review before going live.

Catalog Generation. Each occupation is reconstructed independently by the per-occupation loop of §4.1: starting from the shared baseline prompt, the generator, judge, critic panel, and editor iterate for the fixed budget, after which the parity mapping produces the occupation's best catalog $T^\star$, its tag↔Q&A relation $M^\star$, and the residual sets $U_Q^\star, U_T^\star$. Because the loop is occupation-local, the 132 occupations are processed independently and in parallel. Every run persists a full audit trail: a per-occupation iteration log recording each kept change with its rationale and target dimension, a snapshot of the best tag set and its scores at each iteration, and the parity-mapping outcome. These artifacts are both retained per run and streamed to a shared reviewer workspace as iterations are kept (§5), so the catalog and its rationale are available for inspection while a run is still in progress.

Human QA at Scale. No catalog is deployed without human sign-off, so each generated catalog is seeded into a shared Google Sheet for review. Results were uploaded live during a run into managed per-occupation tabs: *Best* (the current tag set), *Iteration Log* (every kept change with its rationale), *Tricky Parity Cases* (ambiguous legacy mappings flagged for attention), and *Comment Archive* (resolved reviewer comments). The parity tab includes a Verdict dropdown (Keep/Update/Delete) with conditional formatting, and reviewers can inline-edit tag names. Review is not a one-way gate: unresolved reviewer comments are polled from the sheet (via the Drive API) and written to a per-occupation feedback file, which the generator and editor read on the next run. Human feedback, such as renamed tags, deletions, and free-text objections, therefore feeds directly back into the automated loop, closing the gap between manual review and regeneration.

Scale and Coverage. The system has been live across 132 occupations. New occupations are onboarded on demand from a single configuration entry: a per-occupation config file maps each occupation slug to its data-warehouse table names and primary keys, and adding an occupation requires only that entry. *Automotive Detailing*, for example, was added and taken end-to-end: baseline generation, autoresearch loop, parity mapping, and human QA review, within five hours on a single day in May 2026.

6 Evaluation and Operational Lessons

6.1 Catalog Quality vs. Deployment Outcomes

The autoresearch loop produces a catalog whose quality is measured in one space (the $\mathcal{E}$ composite plus the seven critic penalties), and the deployment is measured in another (filter-evaluation outcomes, occupation concentration). Conflating the two muddles attribution: a missing-canonical-tag filter rate of 40.66% in production is not a verdict on the catalog, it is a verdict on the deployment-side seeding step that was supposed to attach canonical tags to existing pro profiles. Likewise, the loop-internal critic penalties say nothing about whether request-time enrichment actually emits the right tags in production. We separate the two strands explicitly in what follows.

6.2 Catalog-Quality Metrics

$\mathcal{E}$ composite distributions. Across the 132-occupation production runs, the per-tag $\mathcal{E}$ composite (out of 15) lands in a relatively

Table 1: Cross-occupation structural-prefix (KV-key) drift on the $n = 14$ occupation cross-section (189 tags), case-folded. The four shared keys confirm partial standardization without a global hierarchy; the eight singletons and case variants are the residual drift targeted by the next autoresearch wave.

Structural prefix (KV-key)	Occupations
`Capability`	9 / 14
`Work type`	9 / 14
`Deliverable`	7 / 14
`Scope`	7 / 14
8 singleton prefixes (each)	1 / 14

narrow band on the best kept catalog per occupation: the bulk of tags score in the 11–14 range, with low-end outliers concentrated on Canonical Coverage (the binary 0-or-2 rubric) for occupations whose legacy categories do not have an obvious single-tag canonical mapping.

Critic-panel score distributions. As designed, the Pro critic accumulates the largest aggregate penalty mass (capped at 2.5 per iteration; Appendix A); the Adversarial critic contributes a sparse but heavy-tailed distribution dominated by the `critical` risk tier; and the Reasoning bonus fires on roughly the iterations where the editor's most recent change was a rationale upgrade.

Cross-occupation tag-name drift. Because each occupation's catalog is generated in isolation, two occupations whose tags should share a name could drift apart. We measure this on a 14-occupation, 189-tag cross-section at two layers:

(i) Pro-facing tag strings (the names shown to pros, e.g., Cabinet installation) all 189 are unique to their occupation. Occupation-local vocabulary is preserved as intended: chess tutoring emits Adult chess tutoring, not a generic Adult that would also fit physical therapy. This is an exact-string check; semantic near-duplicates like Installation vs Installations are a within-occupation Taxonomy-critic concern, not a cross-occupation one.

(ii) Structural category prefixes (the part before the colon in each tag's structured form, e.g., Work type in Work type: Cabinet installation — the axis the matchmaker and pricing model filter on) 12 distinct prefixes across the cohort after case-folding, distributed as shown in Table 1. Four are widely shared: Capability and Work type in 9/14 occupations, Deliverable and Scope in 7/14, confirming partial standardization emerges without a global hierarchy. Two residual drifts are both targets for the next autoresearch wave: (a) eight singleton prefixes, three of which are clearly the same concept under different names (Client preference, Client type, Customer type); and (b) case inconsistency on shared prefixes (e.g., Deliverable vs DELIVERABLE).

Planned dual-evaluator validation. A complementary validation we plan (eval-of-eval) compares the production $\mathcal{E}$ judge against an independently-developed external evaluator (different judge model, different rubric structure, different thresholds) on the same per-occupation tag set, structured around verdict-agreement rate, per-rubric Cohen's κ on the six shared dimensions, and a critic-based qualitative analysis of disagreement tags.

Critic ablation. A single-occupation Parity-removed critic ablation on *Accounting* (Appendix D) re-aggregates the production iteration log under a six-critic configuration. On the observed Accounting run, the Parity critic returned `parity_score` = 7/10 in both kept iterations — above the panel's penalty thresholds — and therefore contributed zero to the critic penalty. Removing Parity from the panel leaves the adjusted-score trajectory and the final best tag set unchanged on this occupation. We read this as evidence that, on occupations where parity-readiness is comfortably above the penalty floor, the Parity critic's seat is insurance against the lower-readiness tail rather than an active per-iteration contributor; characterizing Parity's marginal contribution across the 132-occupation cohort is the next ablation-program step.

6.3 Deployment-Side Metrics

The 14-day production post-launch cohort study (1,840 MVP-enrolled pros, 9,324,798 filter evaluations over 2026-04-28 to 2026-05-12) characterizes the catalog *as deployed*, distinct from the catalog as generated.

Catalog hygiene. Three catalog tags marked `status = 2` (deprecated) were still being emitted by request-time enrichment as *canonical category tags*, driving roughly 486,000 filter events in the window. The implicated tags—*Heavy lifting*, *Bathroom remodel*, *Wallpaper installation or repair*—came from a request-side reverse-mapping table that was never reconciled with the catalog's active-status flag. The autoresearch loop did not surface this finding directly: it produced a per-occupation catalog whose deployment, monitored at the filter-evaluation event level, exposed the request-side / pro-side mismatch. The *surfacing* came from per-event deployment monitoring—any per-event monitoring pipeline against a hierarchical or per-occupation build could have detected the same SQL signal. What per-occupation independence specifically contributes is unambiguous *attribution*: in a unified hierarchy "Heavy lifting" has a defensible existence as a Moving-and-Lifting sub-tag, so the deprecation flag is one signal among many and the diagnosis is "data-quality nuisance"; under per-occupation independence the deprecated tag has no occupation that claims it as canonical, so the request-side emission is unambiguously a request-side / pro-side mismatch and the fix path is mechanical (re-activate the tag or remove from the reverse mapping). The contribution is the diagnostic clarity, not the diagnostic visibility.

Filter outcome distribution. For the same cohort, the strict-match filter outcome distribution is shown in Table 2. The dominant failure mode—missing canonical category tag, 40.66% of all MVP evaluations—is a deployment-stage gap, not a catalog-quality gap: the catalog tag exists, request-side enrichment fires it, but the pro's profile was never seeded with it. This motivates the deployment-side intervention (auto-attach canonical category tags at MVP onboarding) and validates the Parity critic's instinct that catalog completeness without deployment-side wiring is not production coverage.

Occupation concentration. The same cohort surfaced a strong concentration of filter events: *Handyman* alone accounted for roughly 70% of MVP filter events (3.77M of 5.43M filtered evaluations). Four other occupations had structurally high trigger rates: *Roofing/siding*

Table 2: Strict-match filter outcome distribution across 9.3M MVP filter evaluations in the 14-day post-launch window. The dominant failure mode—missing canonical category tag—accounts for 40.66% of all MVP evaluations and is, by itself, larger than the sum of all other filter reasons combined.

Outcome	Filter evaluations	% of MVP evals
PASSED	3,898,008	41.80%
FILTERED: missing canonical category tag	3,791,760	40.66%
FILTERED: missing hybrid-preference data	743,341	7.97%
FILTERED: no canonical tag, no category-level targeting	568,254	6.09%
FILTERED: limitation on RF preference tag	306,531	3.29%
FILTERED: no targeted occupation match	14,740	0.16%
FILTERED: limitation on canonical preference tag	2,164	0.02%
Total	**9,324,798**	**100.00%**

(73.5%), *Home technology* (72.8%), *Doors/windows* (67.0%), *Appliances* (66.5%). This suggests the canonical tag taxonomy for these occupations does not match how pros describe their work. These are concrete targets for the next autoresearch wave, identifiable because per-occupation results are not averaged into a global score.

Directional post-launch signals (extended MVP window). We did an MVP analysis at 8–10 weeks post-launch (June–July 2026) and opened up enrollment from the 1,840-pro cohort above to ~3,900 pros. We got directional, though not causal, evidence consistent with the generated-vs-deployed attribution of §6.1. (i) About 88% of enrolled pros remained enrolled, with attrition concentrated among high-volume providers and attributed to price and lead-mix economics, not catalog semantics. (ii) Of 4,721 "not what I do" lead-declines, 94% came from providers whose recorded preferences already excluded that work. While early, we interpret this as an enforcement gap, not a vocabulary gap. Additionally, a review from our category-management team recommended seven point changes to the deployed taxonomy. (iii) About 35% of newly recorded provider limitations arrive through the LLM refinement flow built on the generated catalog. Where no preference was adopted, around 75% trace to user-experience gaps rather than rejection of a suggested tag (~21% explicit rejections). (iv) Finally, churn and refunds seem to cluster amongst pros who had narrow legacy categories that mapped into broad new-catalog occupations, and providers with no legacy-category crosswalk link received ~35% fewer leads, consistent with the canonical-assignment and parity lessons of §6.4. An outcome-level comparison against legacy forms remains future work.

6.4 Operational Lessons from the Loop

Canonical-tag assignment is structurally different from holistic generation. An earlier version of the pipeline assigned canonical category tags inside the generation prompt. This was migrated to a dedicated mapping-phase LLM call that runs before each mapping batch and injects canonical context. The architectural separation produced both better canonical coverage and more consistent generation (the generator now focuses on holistic tag creation; canonical assignment is a structured decision with its own prompt and examples).

Table 3: Convergence behavior across 132 production occupations. The reported iteration is the one that produced the best kept tag set per occupation (index 0 denotes occupations whose baseline was never beaten); the "adjusted score" is $\mathcal{E}$ composite – critic penalty. Quartiles use the standard linear-interpolation convention (NumPy `percentile` default).

Metric	Value (min / q1 / median / q3 / max)
Iteration index of best tag set	0 / 1 / 2 / 3 / 3
$\mathcal{E}$ composite of best tag set (/ 15)	7.69 / 13.32 / 13.84 / 14.20 / 14.90
Critic penalty on best tag set	0.50 / 2.49 / 3.62 / 4.85 / 9.70
Adjusted score on best tag set (/ 15)	2.34 / 8.63 / 10.17 / 11.41 / 13.96
Tag count on best tag set	2 / 17 / 26 / 44.25 / 136

Compound names are not unions. The canonical assignment rule originally rejected any category whose name contained "and" or "or" (treating it as a union of distinct services). This collapsed canonical coverage on *Electrical* from 10/10 to 1/10 because categories like *Circuit Breaker Panel or Fuse Box* are compound names for a single service, not unions. The refined rule requires *actual RF Q&A disambiguation evidence* before rejecting a union category.

Stopping is budget-driven. The current loop stops at a fixed iteration budget (default 5) rather than at a convergence-based criterion. A confidence-interval criterion is dominated in cost by the multi-trial evaluations it would require; across production runs the kept-iteration count typically saturates well before the budget is exhausted (§6.5). The per-iteration early-skip described in §4.1 step 6 captures most of the available cost saving without committing to a plateau heuristic the implementation does not yet support.

6.5 Convergence Behavior Across Production Occupations

To characterize how often the loop converges within budget, we aggregate the best-iteration summary row for each of the 132 production occupations from the autoresearch tracking logs. Table 3 summarizes the distribution.

Three observations hold across the full 132-occupation cohort. (i) The best kept tag set is reached by iteration 3 for every occupation—36 at iteration 1, 53 at iteration 2, 38 at iteration 3, and 5 that never beat their baseline (index 0)—with none reaching its best after iteration 3, so the kept-iteration count saturates well before the 5-iteration budget. (ii) The $\mathcal{E}$ composite is tightly clustered (median 13.84/15, mean 13.53, interquartile range 13.32–14.20), but the *adjusted* score spreads much wider (median 10.17, IQR 8.63–11.41): 19 of 132 occupations fall in the < 8 band that the production verdict thresholds (§6.2) route to review or flag, almost entirely on critic penalty rather than on a weak judge score. (iii) The critic penalty (range 0.50–9.70, median 3.62) is therefore the dominant source of variance in the adjusted score, consistent with the asymmetric panel design (§A) in which the Adversarial critic alone can contribute up to 2.0 on the `critical` risk tier.

Loop vs. single-shot generation. Iteration 0 of every production run *is* single-shot generation: the shared seed prompt $p^{(0)}$ applied once, with no critics and no editor, so the tracking logs contain the single-pass baseline directly. Across the 132-occupation cohort, the loop lifts the per-occupation E composite from a median of 11.36 at iteration 0 to 13.84 on the best kept set, a median gain of +2.24 points (IQR 1.60–2.64). It improves 119 of 132 occupations (100 by at least one point, 87 by at least two); 5 never beat their baseline, and 8 keep a set with equal-or-lower composite that wins on a smaller critic penalty, since keeps are decided on the adjusted score. This comparison runs under the loop's own judge, so it isolates the value of iterate-and-keep over single-pass generation; whether the judge itself tracks human quality is assessed separately below.

$\mathcal{E}$-only baseline (vanilla LLM-as-judge contrast). To isolate the seven-critic panel's marginal contribution at the verdict-decision boundary, we re-bin all 132 best-iteration rows under an $\mathcal{E}$-only configuration that drops the critic penalty entirely (`critic_penalty` := 0) and applies the production thresholds (`auto_approve` $\geq$ 11, `human_review` $\geq$ 8, `flag` $<$ 8). The two configurations disagree on 84 of 132 occupations (63.6%), and every disagreement is in the same direction—the panel's verdict is the same or stricter, the expected sign because the critic penalty is non-negative. The distributions tell the story: $\mathcal{E}$ alone would `auto_approve` 125 of 132 occupations (94.7%), leaving a human-review queue of only 7; the panel routes just 47 to `auto_approve`, 66 to `human_review`, and 19 to `flag`. The panel's dominant operating mode is therefore to demote tag sets the judge alone would wave through—it is what creates the 85-occupation human-in-the-loop queue the production system relies on. This is a verdict-bucket comparison on the converged best-iteration tag sets, not a re-run of the loop with the panel disabled; the latter (the panel's effect on the kept-iteration *trajectory*) is the ablation scoped in §6.4.

$\mathcal{E} \leftrightarrow$PM-consensus agreement. Table 7 cross-tabulates the model's verdict against the production PM-consensus action over the 4,112 tags that received both, under the verdict map Keep↔auto-approve, Update↔human-review, Delete↔flag. Raw agreement is 77.2% (3,174/4,112), but this is almost entirely a base-rate effect: the model auto-approves 97.3% of tags and reviewers keep 79.1%, so chance agreement is already 77.2% and Cohen's $\kappa \approx 0$. The discrete E verdict does not, on its own, predict which tags a human will reject. We report this negative result since it is the basis for using the $\mathcal{E}$ composite as a ranking signal inside the loop rather than an acceptance gate, and for keeping human review mandatory before any catalog deploys. One caveat is circularity on the legibility rubric, which was recalibrated from the same PM review (§4.2).

7 Discussion

Our contribution is not a single algorithm but a system architecture for reconstructing catalogs in marketplaces that carry substantial legacy structured data, where getting the pro-side schema right has downstream consequences. The composition we defend has three elements: per-occupation independence, the seven-persona critic panel, and a parity-mapping stage. While each module has its own ancestry, what we add is their integration, the scale at which the system runs in production (132 occupations, with new ones onboarded on demand), and the operational lessons that scale has surfaced. We see two clear extension paths as future work. First, the search backend is currently a naive LLM proposal. GEPA-style reflective optimization [1] with execution traces is an upgrade once the critic-panel output is structured enough to serve as a reflection signal. Second, the production feedback loop is currently human-in-the-loop (PM comments flow back via the live sheet). Routing matchmaker-side production failures (eg, 41% missing-canonical-tag filter rate) into the next autoresearch iteration would close the loop between catalog generation and matchmaking outcomes.

Trustworthiness considerations. The cross-family arrangement described in §4.1 mitigates self-reinforcement between the generator (GPT-5-4 family) and the critics, editor, and parity mapper (Claude Sonnet 4.6 family), but two residual trust gaps remain. (i) Within Claude Sonnet 4.6, the editor and the seven critics share a model family; an editor that is biased in a direction the critics also endorse will not be caught by the loop, although it would be caught by a cross-family critic seat. A natural mitigation is to host one critic on a third model family (e.g., a Gemini- or Llama-class model) and treat its disagreement as an explicit signal to the editor. (ii) Critic disagreement is currently silent: the editor sees the aggregate adjusted score, not the per-critic decomposition, so a high-variance verdict (e.g., the Adversarial critic flags a critical failure that no other critic surfaces) is averaged into the same scalar as a low-variance verdict. Surfacing per-critic penalties and a disagreement summary as explicit editor inputs is a direct upgrade to the loop's trustworthiness, and is a planned next iteration.

8 Conclusion

We presented a per-occupation autoresearch system for generating marketplace catalogs, in production at a 132-occupation consumer services marketplace. The contributions we defend are compositional: the unit of optimization (preference-tag catalog within an occupation), the per-occupation parallelism, the seven-persona critic panel with weighted penalties, the legacy-Q&A-to-modern-tag migration via Step-Back abstraction, and the MEMORY_ONLY typing primitive. Production deployment at scale surfaced an operational catalog-hygiene finding (three deprecated tags driving 486K filter events) that a global hierarchical build would have masked.

References

[1] Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2025. GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning. arXiv:2507.19457

[2] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073 [cs.CL]

[3] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

[4] Liran Einav, Chiara Farronato, and Jonathan Levin. 2016. Peer-to-Peer Markets. *Annual Review of Economics* 8 (2016), 615–635.

[5] Yunhan Huang, Klevis Ramo, Andrea Iovine, Melvin Monteiro, Sedat Gokalp, Arjun Bakshi, Hasan Turalic, Arsh Kumar, Jona Neumeier, Ripley Yates, Rejaul Monir, Simon Hartmann, Tushar Manglik, and Mohamed Yakout. 2025. AttributeForge: An Agentic LLM Framework for Automated Product Schema Modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2106–2121. doi:10.18653/v1/2025.emnlp-industry.148

[6] Hamel Husain. 2024. Critique Shadowing: Building Trustworthy LLM-as-Judge Systems. Online essay, https://hamel.dev/blog/posts/llm-judge/. Accessed 2026-06-09.

[7] Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. 2025. TaxoAdapt: Aligning LLM-Based Multidimensional Taxonomy Construction to Evolving Research Corpora. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://aclanthology.org/2025.acl-long.1442/

[8] Andrej Karpathy. 2026. autoresearch: AI agents running research on single-GPU nanochat training automatically. GitHub repository. https://github.com/karpathy/autoresearch Released March 7, 2026.

[9] Nan Li, Bo Kang, and Tijl De Bie. 2025. Building Data-Driven Occupation Taxonomies: A Bottom-Up Multi-Stage Approach via Semantic Clustering and Multi-Agent Collaboration. arXiv:2509.15786 Introduces the CLIMB multi-agent framework for occupation hierarchy construction.

[10] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[11] Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing Instructions and Demonstrations for Multi-Stage Language Model Programs (MIPROv2). In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[12] Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Xu, Subhabrata Mukherjee, et al. 2024. TnT-LLM: Text Mining at Scale with Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*.

[13] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. In *International Conference on Learning Representations (ICLR)*.

[14] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Scoring Details

Tables 4, 5, 6 enumerate the judge scores, critic panel scores, and the set penalty rules applied.

The composite score that drives the keep/discard decision is $a = s - \rho(T) - \sum_k w_k \pi_k(T)$, where s is the E1 set score (Table 4), $\rho(T)$ is the set-level penalty (Table 6), and $\sum_k w_k \pi_k(T)$ is the aggregate critic penalty (Table 5). The pre-critic quantity $\hat{a} = s - \rho(T)$ is floored at 0 and gates whether the critic panel runs.

Table 4: E1 LLM-as-judge dimensions. Each tag is scored on all six rubrics; the per-tag composite is their sum on a $/15$ scale. The set score s is the mean per-tag composite over the tag set T, $s = \frac{1}{|T|} \sum_{\tau \in T} \text{composite}(\tau) \in [0, 15]$.

Rubric (per tag)	Range
Screening vs. Intake	0–3
Tag Legibility	0–3
Preference Variance	0–3
Cross-Category Consistency	0–2
Canonical Coverage	$\{0, 2\}$
Information Loss	0–2
Per-tag composite (sum)	0–15
Set score s (mean over T)	$[0, 15]$

Per-tag verdict bands: auto-approve ($\geq$ 11), human-review (8–10), flag ($<$ 8).

Table 5: Critic panel. Each persona maps its structured verdict to a penalty; the aggregate critic penalty is the sum, $\sum_k w_k \pi_k(T)$. All scores are LLM-assigned on a 0–10 scale unless noted; counts are non-negative integers. The Reasoning critic is the only bonus-eligible persona.

Critic	Penalty rule	Cap / bonus
PM	$0.15 \cdot \max(0, r - 2)$, r=#rejections	first 2 rejections free
Pro	$0.15v + 0.2i$ (v=vetoes, i=irrelevant)	capped at 2.5
Taxonomy	$0.2 \cdot \max(0, 5 - \sigma)$, σ=structural score	—
Adversarial	risk{low 0, med 0.5, high 1.0, crit 2.0} + 0.3·#critical failures	—
Reasoning	< 4: +1.0; < 6: +0.5; ≥ 8: −0.3	bonus −0.3
Coherence	< 5: +1.0; < 7: +0.5	—
Parity	< 3: +1.0; < 5: +0.5	neutral 5 if no data

The aggregate can be slightly negative when the Reasoning bonus exceeds other penalties.

B Worked Example: One Autoresearch Iteration on Accounting

To make the loop concrete, this appendix walks through a single iteration on *Accounting* (37 tags, the smallest production occupation). The walkthrough is reconstructed from the live tracking sheet's *Iteration Log* tab and the per-iteration production snapshots. Numerical values are stated as observed.

Table 6: Set-level penalty $\rho(T)$. Components are summed; $\hat{a} = s - \rho(T)$ is floored at 0. The positive/negative parity terms use the pro-weighted regrettable-coverage ratios r_+, r_- and activate only below a 0.90 coverage target.

Component	Rule	Cap
Tag proliferation	+0.5 per parent tag with > 8 children	uncapped[†]
Parity category gaps	1.0·#fully-unmapped categories	3.0
Positive parity (enroll.)	$10\,(0.90 - r_+)$ if $r_+ < 0.90$	4.0
Negative parity (desel.)	$10\,(0.90 - r_-)$ if $r_- < 0.90$	6.0
$\rho(T)$ total	sum of the above	$\sim [0, 13]$

[†]Implemented guardrail; the current generator emits no tag hierarchy, so this term did not activate in our runs.

Setup. The starting prompt $p_o^{(t)}$ for Accounting is the team's current best prompt as of the run, with prior kept changes already applied. The legacy data feed for Accounting includes the category-to-RF-question schema for the occupation's mapped categories (e.g., *Tax preparation*, *Bookkeeping services*, *Accounting consultations*) and the observed answer distributions per category from the 90-day deselection window.

Step 1 (Generate). The generator (GPT-5-4) emits a candidate tag set $T^{(t)}$ of $\sim$ 37 tags. A representative slice: *Individual tax returns*, *Small-business bookkeeping*, *Quarterly tax filings*, *Audit representation*, *Cash-basis accounting*, *Accrual-basis accounting*, *Payroll services*, *Sales and use tax*, *Multi-state filings*, *Nonprofit accounting*.

Step 2 (Evaluate, $\mathcal{E}$). GPT-5-4-mini scores the set on the six rubrics. Per-rubric means cluster as: Screening vs. Intake $\approx$ 2.7/3, Tag Legibility $\approx$ 2.6/3, Preference Variance $\approx$ 2.4/3, Cross-Category Consistency $\approx$ 1.7/2, Canonical Coverage = 2/2 (every legacy category mapped), Information Loss $\approx$ 1.5/2. Composite $s^{(t)} \approx 12.9/15$. The lowest-scoring rubric is Information Loss, driven by the consolidation of several legacy answer variants under a single tag.

Step 3 (Critique, panel). All seven critics run in parallel on the full 37-tag set, with the Parity critic also seeing the 90-day deselection top-30 for Accounting. Representative aggregated penalties on this iteration:

- PM: 1 ship-block on a tag with awkward phrasing $\Rightarrow$ penalty 0 (under the free-allowance threshold of 2 rejections).
- Pro: 1 veto + 2 irrelevant flags $\Rightarrow \min(0.15 + 0.40,\ 2.5) = 0.55$.
- Taxonomy: structural score 7/10 $\Rightarrow$ penalty 0.
- Adversarial: low risk, no critical failures $\Rightarrow$ penalty 0.
- Reasoning: rationale-quality score 8/10 $\Rightarrow$ bonus −0.3.
- Coherence: 8/10 $\Rightarrow$ penalty 0.
- Parity: 7/10 $\Rightarrow$ penalty 0.

Net critic penalty $\approx$ 0.25. Adjusted score $s_{\text{adj}}^{(t)} \approx 12.65/15$.

Step 4 (Edit). The editor agent (Claude Sonnet 4.6) is shown the lowest-scoring rubric (Information Loss) and the highest-penalty critic (Pro). Its proposed change is a targeted edit to the cross-category consolidation rule in the prompt: *"Distinguish cash-basis vs. accrual-basis bookkeeping at the tag level rather than as composition_rationale; pros self-select differently."* It is a single-paragraph delta against $p_o^{(t)}$, not a rewrite.

Step 5 (Keep / discard). Regenerating with $p_o^{(t+1)}$ produces a candidate set with one new tag (*Cash-basis bookkeeping*) and a refined version of *Accrual-basis accounting*. The new pre-critic $\mathcal{E}$ composite is $\approx 13.2/15$ (a +0.3 delta on Information Loss without regression elsewhere); the new critic penalty drops to ≈ 0.05 (Pro penalty 0.35 from 1 veto + 1 irrelevant flag, down from iter-1's 0.55, with the Reasoning −0.3 bonus retained). New adjusted score $\approx 13.15/15 > 12.65$, so the change is *kept*, and the kept change is logged to the *Iteration Log* tab with the editor's stated rationale.

The walkthrough above is one of several iterations the live sheet records for Accounting.

C Model Verdict vs. Reviewer Action

Table 7 cross-tabulates the model's $\mathcal{E}$ verdict against the reviewer's action for the 4,112 tags that received both.

Table 7: Confusion matrix between the model's E1 verdict (columns) and the human reviewer's action (rows), over 4,112 tags that received both. Categories are aligned as auto_approve ↔ `Keep`, human_review ↔ `Update`, flag ↔ `Delete`.

	Model E1 verdict			
Reviewer	auto_approve	human_review	flag	**Total**
`Keep`	3,164	88	1	3,253
`Update`	292	10	0	302
`Delete`	547	10	0	557
Total	4,003	108	1	4,112

D Critic Ablation: Accounting with the Parity Critic Disabled

This appendix reports a single-occupation critic ablation. The premise of the seven-critic panel is that no single critic carries the loop alone; the Parity critic in particular is the panel's distinctive coverage signal. To probe how much the Parity critic contributes on a representative small occupation we run *Accounting* twice—once in the production seven-critic configuration, once with the Parity critic removed from the panel—and compare the kept-iteration trajectory and the final adjusted score. We choose Accounting because (a) it is the smallest production occupation (37 tags, $\approx$ 15–20 minutes per run) so the ablation is cheap, and (b) it has a non-trivial deselection signal so the Parity critic is informative on it.

Methodology — synthetic ablation from the production iteration log. The ablation re-aggregates the seven per-critic outputs in the production Accounting iteration log under the production seven-critic configuration vs. a six-critic configuration with the Parity entry removed, using the weight formulas in Appendix A. The synthetic ablation reproduces the *score delta* that Parity contributed per iteration, holding the rest of the run trajectory fixed. It does not simulate the editor agent's response to a different adjusted-score trajectory; a full run with Parity disabled would require re-invoking the generation+evaluation+editor pipeline end-to-end on a clean Bedrock environment, and is reserved for the multi-occupation ablation study scoped in §6.4.

Result. On the Accounting run, Parity returned `parity_score` = 7/10 on iteration 1 and 7/10 on iteration 2. Both scores are above the penalty thresholds (penalty fires only at `parity_score` < 3 (+1.0) or < 5 (+0.5); Appendix A row 7), so the Parity critic contributed exactly 0.0 to the critic penalty on both observed iterations. Removing Parity from the panel therefore leaves the production adjusted scores unchanged (12.65 on iteration 1, 13.15 on iteration 2) and the keep/discard trajectory unchanged (13.15 > 12.65 is kept under both configurations). The final best tag set is identical.

Reading the result. On Accounting, in this run, the Parity critic is *redundant* with the existing positive-parity-readiness floor — its panel seat is reserved insurance against occupations where `parity_score` drops below 5 rather than an active contributor to the kept/discard decision on this occupation. This is the negative-result-as-evidence pattern the appendix opens with: a single-occupation ablation cannot characterize the Parity critic's marginal contribution across the 132-occupation cohort, but it does establish that on at least one production occupation the Parity critic earns its seat through the threshold structure (no penalty when parity-readiness is comfortable) rather than through active per-iteration influence. A multi-occupation ablation across occupations with lower `parity_score` (e.g., occupations where the per-occupation deselection cohort is shallow; cf. the activation-rate distribution in §4.4) is the next step in the ablation program and is reserved for the camera-ready follow-up.