One Prompt Does Not Fit All: Self-Meta-Evolve for Personalized Information Extraction

Hongliang Li[1,2]*, Lu Wang[1], Yong Xu[1], Hanyang Chen[2], Zhitao Hou[1], Xiaoting Qin[1], Song Ge[1], Qingwei Lin[1], Dongmei Zhang[1]

[1]Microsoft, [2]Beijing Jiaotong University

Abstract

Large language models (LLMs) are increasingly deployed for enterprise information extraction (IE), where the same document must be reorganized differently for each user. Existing prompt optimization methods, however, rely on a single prompt optimized against a global objective, which is misaligned with the inherent *user heterogeneity* of real workplaces. We formulate enterprise IE as **per-user prompt adaptation under interaction feedback** and propose **Self-Meta-Evolve**, a hierarchical framework that maintains a dedicated prompt for each user and continuously refines it through a dual-loop process: an inner loop that edits structured prompts based on persona-conditioned feedback, and an outer loop that evolves the meta-prompt itself by distilling successful editing patterns. To enable scalable training and evaluation, we release a persona-driven IE benchmark of 292 simulated enterprise users, paired with a reproducible persona-generation pipeline grounded in O*NET occupational taxonomies. On this benchmark, Self-Meta-Evolve achieves a 74.58% success rate, outperforming the strongest prompt-optimization baseline by 13.56 absolute points, and reaches 52.54% within only two iterations. A double-blind human study with twenty real professionals further confirms that prompts adapted by our framework win against static baselines in 71% of pairwise comparisons.

1 Introduction

Large language models (LLMs) are now a core building block of enterprise data pipelines, transforming unstructured artifacts such as emails, meeting transcripts and project reports into structured representations that support downstream analytics and decision-making (Pang et al., 2023; Xu et al., 2024; Liang et al., 2026). In production, prompts have become the de-facto interface for controlling these systems: many platforms invoke a closed-source LLM via a single carefully engineered prompt and treat the model as a configurable extractor (Xue et al., 2024; Shuang et al., 2025; Fuente et al., 2025).

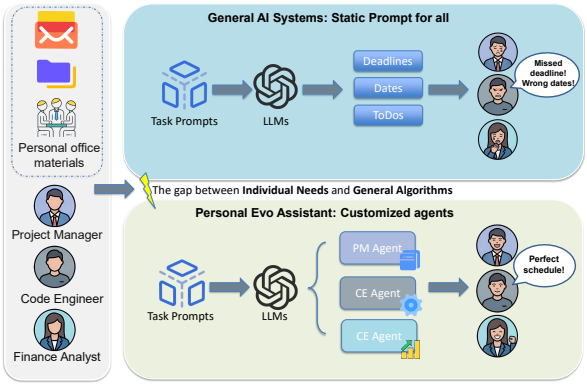


Figure 1: Static prompts fail under user-specific extraction objectives. Conventional systems apply a shared prompt to all users, often producing outputs that do not match heterogeneous role-specific preferences. Our framework maintains a dedicated prompt for each user and updates it from interaction feedback, enabling personalized extraction over time.

This prompt-based paradigm implicitly assumes that one prompt can adequately specify the extraction objective for the entire user population. In practice, however, enterprise users are profoundly heterogeneous (Chen et al., 2024; Shi et al., 2025; Xu et al., 2025): as illustrated in Figure 1, a project manager looking at a project status email cares about actionable deadlines and ownership, whereas a financial analyst reading the same email needs granular budget entities and risk indicators. Our pilot survey of 50 internal users found that 72% routinely post-edit generic LLM outputs to fit their role-specific standards. Recent enterprise-NLP studies corroborate this gap: when LLM agents are evaluated in realistic workplace sandboxes (Vishwakarma et al., 2025) or applied to het-

*Work conducted during Hongliang Li's internship at Microsoft.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3392–3407
November 6-10, 2026 ©2026 Association for Computational Linguistics

erogeneous enterprise data (Choubey et al., 2025), role-specific output requirements become the dominant source of failure. Organizational research provides a theoretical lens for this gap: the Job Crafting framework (Wrzesniewski and Dutton, 2001) and Work Design Theory (Hackman and Oldham, 1976; Parker et al., 2017) argue that effective tools must align with the dynamic, role-specific demands that employees actively shape—a property that static, global prompts cannot satisfy.

Existing prompt optimization methods, including search-based (Zhou et al., 2023; Yang et al., 2024), gradient-inspired (Pryzant et al., 2023), evolutionary (Xu et al., 2022; Guo et al., 2024; Fernando et al., 2024), and multi-agent variants (Zhang et al., 2026), share the same global assumption: a single prompt is optimized against a fixed dataset and shared objective. This is fundamentally misaligned with user-centric IE, where the "correct" extraction is a function of who is reading it (Guan et al., 2025).

We instead formulate enterprise IE as **per-user prompt adaptation under interaction feedback**: each user owns a dedicated prompt that evolves over time, driven by user-specific critique signals. Two observations motivate our design. First, modern LLMs exhibit strong role-playing fidelity, enabling scalable simulation of diverse user personas and their feedback (Park et al., 2023; Tseng et al., 2024; Xiao et al., 2024; Wang et al., 2024; Tamoyan et al., 2025). Second, prompt optimization decomposes naturally into a hierarchy: low-level edits to a task prompt, and a high-level meta-strategy that decides *how* to edit (Deng et al., 2022; Fernando et al., 2024; Zhang et al., 2026).

Building on these observations, we propose **Self-Meta-Evolve**, a hierarchical framework whose *inner loop* performs persona-conditioned edits to a structured extraction prompt, while an *outer loop* evolves the meta-prompt that controls these edits by distilling successful trajectories. To support scalable yet faithful training, we additionally curate a benchmark of 292 enterprise personas, generated through a reproducible pipeline seeded with O*NET occupational descriptors (Park et al., 2023; Wang et al., 2023), and validated against real professionals.

Our intended setting is persistent multi-user personalization: editing experience accumulated across users can improve subsequent adaptations. A direct prompt edit remains a practical option for a single user with a simple requirement. Once adapted, a prompt can be reused for subsequent documents without repeating optimization; deployment then requires one extraction call.

We summarize our contributions as follows:

- **Problem formulation.** We formalize enterprise IE as per-user prompt adaptation under interaction feedback, departing from the static, global prompt optimization paradigm.
- **Self-Meta-Evolve framework.** We propose a hierarchical framework that combines per-user prompt editing (inner loop) with shared meta-strategy evolution (outer loop), and analyze how each component contributes to convergence speed and cross-domain generalization.
- **Persona-driven benchmark and human validation.** We release a benchmark of 292 reproducible enterprise personas plus two specialized test sets (STEM, Humanities), and validate the simulation through a double-blind study with twenty real professionals (Cohen's κ=0.71). Self-Meta-Evolve improves success rate by 13.56 absolute points over the strongest baseline and wins 71% of pairwise comparisons against a static baseline judged by real users.

2 Related Work

Prompt optimization. Search-based methods such as APE (Zhou et al., 2023) and OPRO (Yang et al., 2024) treat LLMs as black-box optimizers that propose and rank candidate prompts; gradient-inspired methods like ProTeGi (Pryzant et al., 2023) use textual critiques as pseudo-gradients. Evolutionary variants (Xu et al., 2022; Guo et al., 2024; Chen et al., 2023; Fernando et al., 2024) apply genetic operators over populations of prompts, and Promptbreeder additionally evolves the mutation prompt itself. Recent work explores meta-learned (Choi et al., 2026), Pareto-optimal (Zhao et al., 2025a), and multi-agent (Zhang et al., 2026) prompt search. All of these optimize a *single* prompt against a *global* objective. Our framework departs from this assumption: prompts are user-specific, and the optimization signal is a persona-conditioned critique rather than ground-truth labels.

LLM-based simulation, judging, and self-improvement. LLMs have been used to simulate users (Park et al., 2023; Xiao et al., 2024;

Tamoyan et al., 2025) and to judge generations on per-instance criteria (Zheng et al., 2023; Liu et al., 2023; Fu et al., 2024; Chirkova et al., 2026). Dou et al. (2025) systematically study whether LLM user-simulators are reliable proxies for multi-turn evaluation and find moderate but non-trivial fidelity, motivating cross-model judge checks. Self-improving systems use LLMs to mutate code, heuristics, or prompts via reflective evaluation (Ye et al., 2024; Liu et al., 2024; Wang et al., 2026). Our work differs in that simulation is *the substrate of personalization*, not just an evaluation tool: AI-User feedback drives per-user adaptation, and the meta-prompt is evolved across users rather than within a single optimization run.

Personalized NLP and role-playing agents. Personalization has been studied in dialogue (Zhang et al., 2018; Madotto et al., 2019), classification (Flek, 2020), fine-grained text generation (Alhafni et al., 2024), and recently with LaMP-style benchmarks (Salemi et al., 2024; Salemi and Zamani, 2025; Salemi et al., 2025). Surveys on persona/role-playing agents (Tseng et al., 2024; Chen et al., 2024, 2025a; Xu et al., 2025; Guan et al., 2025) highlight the gap between role-playing fidelity and downstream task utility, and recent benchmarks evaluate personalization in conversational assistants (Zhao et al., 2025b; Mok et al., 2025) and few-shot personalization with mis-aligned responses (Kim and Yang, 2025). Role-playing capabilities have been explicitly elicited in RoleLLM (Wang et al., 2024). At the same time, Kim et al. (2025) and Luz de Araujo et al. (2026) point out that personas can have both benefits and drawbacks, and their effectiveness may decline during longer interactions. We account for these issues through an outer-loop reset mechanism (Section 6). Our work introduces personalization into enterprise IE, a commercially important yet relatively underexplored setting, and uses occupational taxonomies as the basis for persona construction to support reproducibility.

LLM-based information extraction and synthetic data. LLM-driven IE has been explored for relation (Xue et al., 2024; Jiang et al., 2024), document-level (Liang et al., 2026), event (Shuang et al., 2025), and zero-shot multi-agent (Lu et al., 2025) settings; surveys provide broader coverage (Xu et al., 2024). Synthetic data has emerged as a key enabler for low-resource and zero-shot IE (Fuente et al., 2025; Ziegler et al., 2025). Closest in spirit, personalized meeting summarization systems (Chen et al., 2025b; Kirstein et al., 2025) adapt to user corrections, but operate at the document-summary granularity rather than at the prompt-program level we target. These works optimize a single prompt against gold annotations or fixed user histories. We complement this line by studying *user-relative* correctness, which is the dominant regime in enterprise deployments where "ground truth" is role-dependent (Vishwakarma et al., 2025; Choubey et al., 2025).

3 Problem Setup

3.1 Per-User Prompt Adaptation

Let $\mathcal{U}$ denote a distribution over enterprise user personas. Each persona $u \sim P_{\mathcal{U}}$ encodes a role description, a document distribution $\mathcal{D}_u$, and an information preference profile. Given an extraction prompt $p \in \mathcal{P}$ and document $x \sim \mathcal{D}_u$, an LLM produces a structured output $y = f_{\text{LLM}}(x, p)$.

Unlike traditional IE benchmarks that assume static gold annotations, enterprise systems evaluate quality from a *user-centric* perspective: the same y may be acceptable for one user and incorrect for another. We model this by a persona-conditioned feedback function $F_\phi(u, x, y)$ that returns structured critiques. The optimization objective is to find a per-user prompt that minimizes the expected feedback-induced loss:

$$p_u^* = \arg\min_{p \in \mathcal{P}} \mathbb{E}_{x \sim \mathcal{D}_u} \left[\mathcal{L}(p; u, x) \right], \quad (1)$$

$$\mathcal{L}(p; u, x) = g\left(F_\phi(u, x, f_{\text{LLM}}(x, p)) \right), \quad (2)$$

where $g(\cdot)$ converts discrete feedback into a scalar (Section 3.2).

3.2 AI User Feedback Model

We instantiate F_ϕ with an **AI User** module that role-plays u via LLM prompting and emits structured feedback $F_\phi(u, x, y) = \{(a_i, t_i)\}_{i=1}^m$, where a_i is an extracted artifact and $t_i \in \{\text{revise}, \text{delete}\}$ specifies the action that the user would take. The scalar loss is

$$\mathcal{L} = \lambda_r N_{\text{revise}} + \lambda_d N_{\text{delete}}. \quad (3)$$

Choosing λ_r and λ_d. The weights reflect the relative cost of *editing* a returned artifact versus *discarding* it entirely. We do not set them arbitrarily: in a pilot annotation study, five enterprise users rated the perceived effort of the two actions across

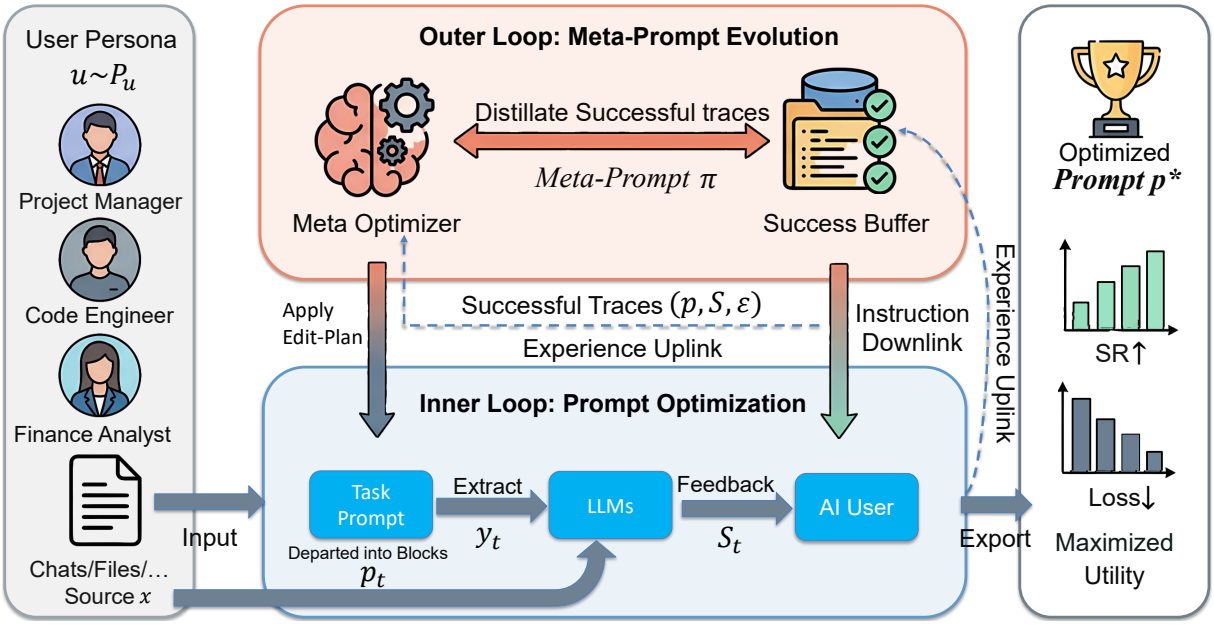


Figure 2: Architecture of Self-Meta-Evolve. The **inner loop** iteratively refines a user's structured extraction prompt based on AI-User feedback, while the **outer loop** evolves the meta-prompt by distilling successful editing trajectories across users.

30 sampled cases on a 1–5 Likert scale, yielding mean ratios of 2.04 ± 0.41 in favor of deletion. We accordingly set $\lambda_d = 0.2 = 2\lambda_r$ and verify in Section 6.7 that final rankings are stable under perturbations of this ratio in $[1.5, 2.5]$.

Loss threshold τ. A run is *successful* if its deployed loss falls below τ. We pick τ as the median deployed loss of the strongest baseline (ProTeGi) on a held-out development set of 50 personas, which corresponds to a normalized $\tau = 0.75$. This calibration ensures that "success" represents a strict improvement over the prior state of the art, and is reused identically across all methods.

4 Persona-Driven Enterprise Benchmark

A central concern for any user-centric IE benchmark is whether the personas faithfully cover real workplace heterogeneity. We therefore describe the construction pipeline in detail (Figure-style overview in Appendix B) and validate it against real professionals (Section 6.9).

4.1 Persona Generation Pipeline

The pipeline is fully reproducible from public seeds and proceeds in four stages.

Stage 1: Seed roles from O*NET. We start from the public U.S. O*NET occupational database (release 28.0), which provides standardized job titles, task statements, and tools-and-technology lists for ∼1,000 occupations. We filter to enterprise-relevant occupations using the SOC major groups 11 (Management), 13 (Business and Financial), 15 (Computer and Mathematical), 17 (Engineering), 19 (Life/Physical/Social Sciences), 23 (Legal), 25 (Education), and 27 (Arts/Media). This yields 412 candidate seed roles.

Stage 2: Persona expansion. For each seed role, we prompt an LLM (GPT-5.1) to expand the O*NET descriptor into a structured persona JSON with three fields: (i) *role description*, including background, seniority, and primary objectives; (ii) *document distribution*, describing the mix of artifact types such as email, chat, report, and ticket, as well as the relevant topical domains; and (iii) *information preference profile*, which contains a list of 3–6 declarative preferences, for example, "prioritize tasks with explicit owners and deadlines over status descriptions." The expansion prompt and an example showing the transformation from a seed role to a persona are provided in Appendix A. We generate 400 candidate personas in this stage.

Stage 3: Quality control. We apply three filters to obtain the final pool: (a) deduplication via embedding cosine similarity (> 0.92 collapses to one), removing 47 near-duplicates; (b) consistency check, where an LLM judge scores the persona on internal coherence (1–5) and we drop scores < 4, removing 38 cases; (c) preference specificity, re-

quiring at least three actionable preferences, removing 23 cases. The remaining **292 personas** form our main pool. This pipeline transparently explains the 400→292 reduction that earlier iterations of this work were criticized for under-documenting.

Stage 4: Document synthesis. For each persona, we synthesize 8–12 enterprise artifacts (emails, chat logs, status reports) conditioned on the persona's document distribution, using a separate generation prompt that injects realistic timestamps, names, and project references. Crucially, the document generator is conditioned only on the *role* and *topical domain*, not on the preference profile, so that documents do not leak the gold preference signal.

For example, a Patent Counsel persona receives fictional invention disclosures, internal patent-review emails, IP-committee meeting notes, and licensing memos, rather than real patents or corporate documents. The generation prompt excludes identifiable names, filing numbers, and jurisdictions.

4.2 Splits and Test Sets

We partition the 292 personas by persona ID into 60/20/20 train/dev/eval splits with a fixed seed. To stress cross-domain generalization, we additionally curate two specialized test sets disjoint from the main pool: **STEM-Personas** ($N = 50$, drawn from SOC 15/17/19) and **Humanities-Personas** ($N = 50$, drawn from SOC 25/27 plus librarian/editor occupations). Detailed distributions are in Appendix B.

Algorithm 1 Self-Meta-Evolve

Init: prompt p_0, meta-prompt π, buffer $\mathcal{B} \leftarrow \emptyset$
for $t = 0$ to $T - 1$ **do**
  sample persona u and document x
  $y_t \leftarrow f_{\text{LLM}}(x, p_t)$
  $\mathcal{S}_t \leftarrow F_\phi(u, x, y_t)$
  $E_t \leftarrow f_{\text{LLM}}(p_t, \mathcal{S}_t; \pi)$
  $p_{t+1} \leftarrow \text{ApplyEdits}(p_t, E_t)$
  if $\mathcal{L}(p_{t+1}) < \mathcal{L}(p_t)$ **then**
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{(p_t, \mathcal{S}_t, E_t)\}$
  end if
  if $|\mathcal{B}| \bmod K = 0$ **then**
    $\pi \leftarrow f_{\text{Meta}}(\pi, \mathcal{B})$
  end if
end for

4.3 Data Statement and Release

All documents are LLM-synthesized; no real enterprise data is used, removing privacy risk. Personas are derived from a public taxonomy. Upon acceptance we release the persona pool, generated documents, AI-User prompt, and pipeline scripts under CC-BY 4.0 to enable replication. Limitations of the synthetic regime are discussed in the Limitations section.

5 Self-Meta-Evolve Framework

Our framework (Figure 2) performs per-user prompt adaptation through a bi-level process. The inner loop edits a user's prompt from current-interaction feedback, while the outer loop improves a shared meta-strategy by distilling successful update patterns across users. Pseudo-code is given in 1.

5.1 Structured Prompt Representation

Rather than treating prompts as unstructured text, we represent each extraction prompt as a JSON-like object whose keys are functional sections: task description, extraction instructions, output schema, demonstrations, and constraint rules. Each section is identified by a stable `point_id`. This decomposition allows the optimizer to target individual points rather than rewriting the whole prompt, yielding more stable and interpretable updates. The initial prompt is in Appendix D.

5.2 Inner Loop: Persona-Conditioned Editing

At step t, a persona u and document x are sampled. The current prompt p_t produces $y_t = f_{\text{LLM}}(x, p_t)$, the AI User returns feedback $\mathcal{S}_t = F_\phi(u, x, y_t)$, and an LLM editing agent guided by the meta-prompt π proposes an edit plan $E_t = f_{\text{LLM}}(p_t, \mathcal{S}_t; \pi)$ targeting specific (`section`, `point_id`) pairs. The next prompt is

$$p_{t+1} = \text{ApplyEdits}(p_t, E_t). \tag{4}$$

If $\mathcal{L}(p_{t+1}; u, x) < \mathcal{L}(p_t; u, x)$, the trace $(p_t, \mathcal{S}_t, E_t)$ is appended to a success buffer $\mathcal{B}$.

Worked example. Consider a persona "DevOps Lead" whose preference profile demands *actionable on-call follow-ups*. The initial prompt extracts both follow-ups and team announcements; AI-User feedback flags announcements with `delete`. The inner loop localizes the failure to point `instr.scope` ("extract all team-relevant updates"),

and the editing agent proposes a `patch_logic` action that narrows the scope to "items with explicit owner and deadline." Loss drops from 1.4 to 0.6, and the trace enters $\mathcal{B}$.

5.3 Outer Loop: Meta-Prompt Evolution

While the inner loop directly improves *prompts*, the outer loop refines the *strategy* for generating edits. Once $|\mathcal{B}|$ reaches a threshold K, an LLM acting as a meta-optimizer reads the accumulated successful traces and rewrites the meta-prompt:

$$\pi' = f_{\text{Meta}}(\pi, \mathcal{B}). \tag{5}$$

Intuitively, this process captures editing patterns that generalize across users. For example, when feedback indicates over-extraction, the model may prefer applying `add_constraint` to `instr.scope` instead of rewriting demonstrations. These patterns are incorporated into the meta-prompt so that later inner-loop iterations can adapt to *unseen* personas with fewer steps. We therefore assess the quality of the meta-prompt by its adaptation efficiency on held-out personas, which is measured using SR@t (Section 6).

6 Experiments

We address three research questions. **RQ1.** Does Self-Meta-Evolve outperform state-of-the-art prompt optimizers on per-user IE? **RQ2.** Does the meta-evolutionary mechanism improve early-stage convergence and cross-domain robustness? **RQ3.** How does each component contribute, and how does the framework scale with the number of personas?

6.1 Experimental Setup

Datasets. We use the 292-persona main benchmark (60/20/20 split) and the disjoint STEM/Humanities test sets ($N = 50$ each).

Baselines. We compare against: (1) gradient-inspired ProTeGi (Pryzant et al., 2023); (2) search-based APE (Zhou et al., 2023) and OPRO (Yang et al., 2024); (3) evolutionary EvoPrompt (Chen et al., 2023); (4) optimization-specialized Bandit-UCB (Auer et al., 2002), MetaSPO (Choi et al., 2026), and Pareto Prompt (Zhao et al., 2025a).

- **ProTeGi** (Pryzant et al., 2023): textual gradient via LLM-generated critiques, applied with beam width 4.
- **OPRO** (Yang et al., 2024): meta-prompt aggregating top-4 historical prompts; 4 candidates generated per step.
- **EvoPrompt** (Chen et al., 2023): GA-style evolution with population size 4 and standard NL crossover/mutation.
- **Bandit-UCB** (Auer et al., 2002): each prompt section is an arm; UCB1 balances exploration/exploitation.
- **APE** (Zhou et al., 2023): 10 candidate prompts, dev-set selection.
- **MetaSPO** (Choi et al., 2026): meta-system prompt optimization following the original recipe.
- **Pareto Prompt** (Zhao et al., 2025a): dual-objective (accuracy + persona compliance) Pareto front maintenance.

Metrics. **Success Rate (SR)** is the fraction of personas whose deployed loss is no greater than τ (Section 3). **Mean Loss (ML)** averages the deployed loss over personas. **SR@t** is the cumulative success rate within the first t inner-loop iterations and captures convergence speed.

Models. All methods use the same backbone (GPT-5.1) for both the extractor and the AI User; we additionally validate cross-model robustness in Appendix E.

6.2 Main Results (RQ1)

Table 1 reports the comparison on the held-out evaluation set. Self-Meta-Evolve achieves **74.58% SR**, outperforming the strongest baseline ProTeGi by 13.56 absolute points, and reduces ML from 0.7542 to 0.5122 (-32%). Notably, our method's SR@2 (0.5254) already surpasses the *final* SR of OPRO, EvoPrompt, MetaSPO, APE and Pareto Prompt, confirming that the evolved meta-prompt produces higher-quality edits per step.

A paired bootstrap over matched per-persona success indicators gives $p = 0.0038$ against ProTeGi and $p = 0.0009$ against Self-Frozen, for SR gains of 13.56 and 16.95 percentage points, respectively.

Robustness across backbones. Table 2 reports the additional held-out evaluation with five backbones. Self-Meta-Evolve improves over ProTeGi by 11.87–15.25 percentage points in all five cases.

Method	SR↑	ML↓	SR@2	SR@4	SR@8
ProTeGi	0.610	0.754	0.390	0.475	0.542
OPRO	0.458	1.124	0.271	0.339	0.424
EvoPrompt	0.407	1.286	0.220	0.305	0.356
Bandit-UCB	0.542	0.941	0.322	0.424	0.492
APE	0.339	1.423	0.203	0.271	0.305
MetaSPO	0.373	1.331	0.237	0.305	0.339
Pareto Prompt	0.322	1.458	0.186	0.254	0.288
Self-Frozen (ours)	0.576	0.865	0.407	0.492	0.525
Self-Meta-Evolve	**0.746**	**0.512**	**0.525**	**0.610**	**0.678**

Table 1: Main comparison on the held-out evaluation set ($N = 59$). SR@t denotes cumulative success rate at step t. Self-Frozen disables the outer loop.

Backbone	ProTeGi	Ours	Δ
GPT-5.5	64.41	77.97	13.56
GPT-5.1	61.02	74.58	13.56
GPT-4.1	52.54	64.41	11.87
Claude-Opus-4.8	62.71	76.27	13.56
Claude-3.7-Sonnet	54.24	69.49	15.25

Table 2: Additional backbone evaluation: SR (%) on the held-out set ($N = 59$); Δ is the gain in percentage points.

This complements the independent-judge checks in Appendix E. Additional public-document and shared-objective experiments are reported in Appendix I.

6.3 Convergence and Efficiency (RQ2)

Figure 3 traces optimization curves over 20 iterations. Self-Meta-Evolve exhibits a markedly steeper trajectory: gradient-based ProTeGi plateaus near t=10, while our outer loop continues to lift the ceiling by injecting newly distilled editing patterns into π.

Table 3 reports API cost. While the outer loop adds tokens per iteration, the framework remains the most efficient *per percentage point of SR gained* among non-trivial optimizers.

6.4 Cross-Persona Generalization (RQ2)

We evaluate transfer onto STEM and Humanities personas (Table 4). Most baselines suffer 8.0–16.0 point degradation moving from STEM to Humanities, where preferences are more contextual and less tool-specific. Self-Meta-Evolve narrows the gap to **4.0 points**, indicating that the evolved meta-prompt has internalized cross-domain editing principles rather than overfitting to STEM-style errors.

Method	Tok/Iter↓	Total (M)↓	Tok/%SR↓
ProTeGi	6.5K	0.195	3.19K
OPRO	15.8K	0.474	10.35K
EvoPrompt	18.5K	0.555	13.64K
APE	4.8K	0.144	4.24K
Self-Frozen	8.2K	0.246	4.27K
Self-Meta-Evolve	**12.4K**	**0.372**	**4.98K**

Table 3: Efficiency: API cost per iteration and per absolute %SR over 30 iterations.

Method	STEM		Hum.		Δ
	SR	ML	SR	ML	↓
ProTeGi	0.620	0.742	0.540	0.915	8.0
OPRO	0.480	1.095	0.340	1.380	14.0
Bandit-UCB	0.560	0.960	0.420	1.225	14.0
APE	0.380	1.435	0.220	1.760	16.0
Self-Meta-Evolve	**0.760**	**0.492**	**0.720**	**0.574**	**4.0**

Table 4: Cross-persona generalization ($N = 50$ per group). Δ: drop from STEM to Humanities.

6.5 Ablation Study (RQ3)

Table 5 isolates each component. Removing the outer loop causes the largest SR drop (74.6 → 57.6), confirming that meta-evolution is the principal driver of late-stage gains. The success buffer and the structured edit-plan format both contribute to early-step efficiency, validating our hypothesis that experience replay and plan-based editing are key to sample-efficient prompt search.

6.6 Persona Scaling (RQ3)

How many personas does meta-evolution actually need? We rerun the framework with persona pools of size $\{50, 100, 150, 200, 250, 292\}$ while keeping the eval set fixed (Table 6). SR rises sharply from 50 to 150 (+15.7), then enters a regime of diminishing returns past 200. This indicates that 200–250 personas are sufficient for stable meta-evolution, and our 292-persona pool sits comfortably above the saturation point.

6.7 Hyperparameter Sensitivity

We sweep the loss weights (λ_r, λ_d) to verify that our pilot-derived choice (0.1, 0.2) is not cherry-picked. Table 7 shows that final method ranking is preserved across all configurations within $\lambda_d/\lambda_r \in [1.5, 2.5]$, and SR varies by less than 2 absolute points. This robustness directly addresses prior concerns that the 2:1 ratio was arbitrary.

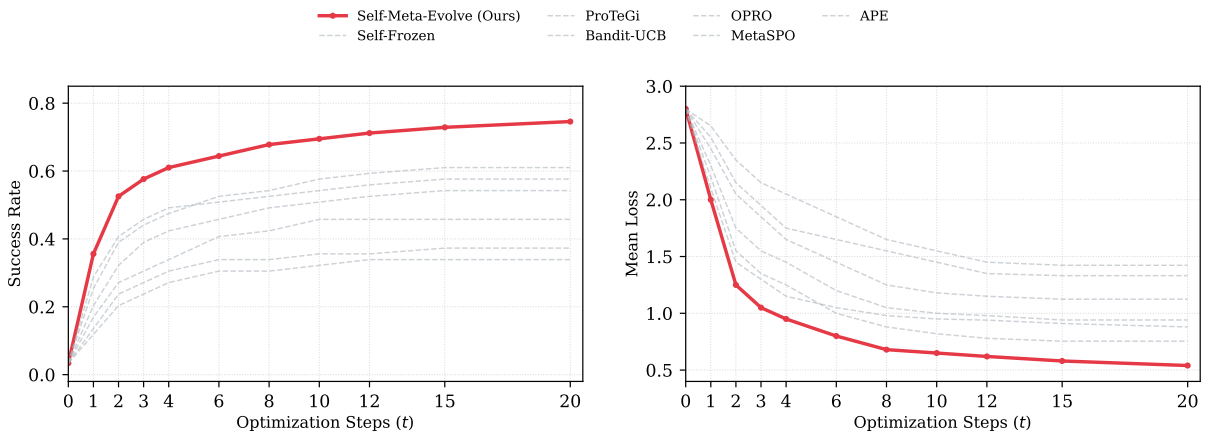


Figure 3: Optimization trajectories. Self-Meta-Evolve maintains higher SR and lower loss across 20 steps. Markers at $t = 2, 4, 8$ correspond to Table 1.

Variant	ML↓	SR↑	SR@2	SR@4	SR@8
Full	**0.512**	**0.746**	**0.525**	**0.610**	**0.678**
− Outer Loop	0.865	0.576	0.407	0.492	0.525
− Success Buf.	0.685	0.661	0.458	0.542	0.593
− Edit Plan	0.621	0.695	0.492	0.576	0.627

Table 5: Ablation on the held-out evaluation set.

#Personas	50	100	150	200	250	292
SR	0.542	0.627	0.699	0.729	0.742	0.746
ML	0.901	0.732	0.611	0.548	0.524	0.512

Table 6: Persona scaling. Returns saturate around 200–250 personas; 292 sits above the knee.

(λ_r, λ_d)	(0.1,0.1)	(0.1,0.15)	(0.1,0.2)	(0.1,0.25)
SR (Ours)	0.737	0.741	**0.746**	0.744
SR (ProTeGi)	0.604	0.607	0.610	0.609

Table 7: Sensitivity to feedback weights. Rankings and absolute SR are stable in the studied range.

6.8 Qualitative Error Analysis

To complement aggregate metrics, we manually classified the 25.4% of personas where Self-Meta-Evolve fails to reach τ within 20 iterations. Three clusters emerge (Appendix F): (i) **Preference conflicts** (43%): mutually inconsistent preferences within a persona that no single prompt can satisfy; (ii) **Domain knowledge gaps** (31%): the LLM lacks the niche terminology to recognize relevant artifacts (e.g., legal-discovery jargon); (iii) **Sparse feedback** (26%): documents containing few persona-relevant items, leading to noisy AI-User signals. The first two are intrinsic limits of any prompt-only adaptation method, and motivate retrieval-augmented extensions discussed in Limitations.

6.9 Human Evaluation

We ran two studies to validate (a) that synthetic personas mirror real workplaces and (b) that prompts adapted by our framework are preferred by real users.

Persona realism. Five evaluators (CS/Linguistics PhD students, unaffiliated) rated 100 mixed samples (50 synthetic + 50 real-world LinkedIn-derived profiles) on four 5-point Likert dimensions in a double-blind setup. As shown in Table 8, synthetic personas score within 0.06 of real ones on every dimension, with no significant differences ($p > 0.5$). Inter-annotator agreement is high (Cohen's $\kappa = 0.71$, Fleiss' $\kappa = 0.68$), addressing prior reviewer concerns about IAA.

These comparison profiles are structured representations of public occupation descriptions and generic role summaries, rather than individual LinkedIn users' profiles. They are used only in the persona-realism study and are excluded from all benchmark splits and optimization loops.

Pairwise preference with real users. We additionally recruited **20 real professionals** (5 each from engineering, product/PM, finance/legal, and editorial/research roles) and showed each of them 10 paired extractions: one from a static baseline prompt and one from the Self-Meta-Evolve-adapted prompt for their role, blinded and order-randomized. Across 200 pairwise judgments, our adapted prompts win 71.0%, lose 19.0%, and tie

Dimension	Real	Synth.	Δ	p
Role consistency	4.78	4.74	-0.04	0.68
Technical depth	4.62	4.65	$+0.03$	0.74
Format realism	4.88	4.82	-0.06	0.52
Language naturalness	4.70	4.68	-0.02	0.81
Overall	**4.75**	**4.72**	$\mathbf{-0.03}$	**0.69**

Table 8: Persona realism (5 evaluators, 100 samples; independent t-test). Cohen's $\kappa = 0.71$.

10.0% (binomial test $p < 10^{-5}$). This directly confirms that the gains from synthetic-on-synthetic optimization transfer to real users.

AI-User feedback authenticity. On 100 sampled feedback instances, evaluators labelled 78% as "human-like" or "uncertain" and only 22% as suspected AI. Distribution is plotted in Figure 9. This is consistent with recent findings that LLM user-simulators can serve as moderate-fidelity proxies for human evaluators (Dou et al., 2025; Tamoyan et al., 2025; Chirkova et al., 2026).

7 Conclusion

We formulated enterprise IE as per-user prompt adaptation under interaction feedback and proposed Self-Meta-Evolve, a hierarchical framework that combines per-user inner-loop editing with cross-user outer-loop meta-evolution. Together with a reproducible 292-persona benchmark grounded in O*NET and validated by twenty real professionals, our framework lifts SR by 13.56 points over the strongest baseline, narrows the STEM–Humanities gap to 4.0 points, and wins 71% of pairwise judgments from real users. Future directions include retrieval-augmented prompts to address domain-knowledge failures and multi-modal extraction for richer enterprise artifacts.

Limitations

Reliance on AI-user feedback. Our training-time feedback is generated by an LLM-based AI User rather than humans. Although the pairwise study with 20 professionals shows that the resulting prompts transfer effectively (71% win rate), there remains a risk that optimization favors patterns that are easier for LLM judges to assess rather than those preferred by real users (Dou et al., 2025; Kim et al., 2025). Cross-model robustness checks (Appendix E) reduce but do not fully remove this concern. Future deployments should incorporate real-user feedback to continually recalibrate the adaptation process.

Coverage of personas and environments. The 292 personas are derived from U.S. O*NET occupations and primarily reflect English enterprise settings. Different organizational cultures, languages, and domain-specific environments may exhibit substantially different information preferences. Extending the pipeline with localized taxonomies and datasets remains future work.

Limits of prompt adaptation alone. As shown in Section 6.8, many remaining failures arise from missing domain knowledge rather than preference mismatch. Prompt adaptation can improve alignment with user preferences, but it cannot supply knowledge absent from the underlying model. Retrieval- or tool-augmented variants may help address this limitation.

References

Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. 2024. Personalized text generation with fine-grained linguistic control. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 88–101, St. Julians, Malta. Association for Computational Linguistics.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256.

Angelica Chen, David Dohan, and David So. 2023. Evoprompting: Language models for code-level neural architecture search. *Advances in neural information processing systems*, 36:7787–7817.

Chaoran Chen, Bingsheng Yao, Ruishi Zou, Wenyue Hua, Weimin Lyu, Toby Jia-Jun Li, and Dakuo Wang. 2025a. Towards a design guideline for RPA evaluation: A survey of large language model-based role-playing agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18229–18268, Vienna, Austria. Association for Computational Linguistics.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, and 1 others. 2024. From persona to personalization: A survey on role-playing language agents. *arXiv preprint arXiv:2404.18231*.

Zheng Chen, Jiang Futian, Yue Deng, Changyang He, and Bo Li. 2025b. Meetalk: Retrieval-augmented and adaptively personalized meeting summarization with knowledge learning from user corrections. In

Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM), pages 94–110, Vienna, Austria. Association for Computational Linguistics.

Nadezhda Chirkova, Tunde Oluwaseyi Ajayi, Seth Aycock, Zain Muhammad Mujahid, Vladana Perlić, Ekaterina Borisova, and Markarit Vartampetian. 2026. LLM-as-a-qualitative-judge: automating error analysis in natural language generation. In *Proceedings of the First Workshop on Multilingual Multicultural Evaluation*, pages 99–132, Rabat, Morocco. Association for Computational Linguistics.

Yumin Choi, Jinheon Baek, and Sung Ju Hwang. 2026. System prompt optimization with meta-learning. *Advances in Neural Information Processing Systems*, 38:55451–55488.

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. 2025. Benchmarking deep search over heterogeneous enterprise data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 501–517, Suzhou (China). Association for Computational Linguistics.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. 2025. Simulatorarena: Are user simulators reliable proxies for multi-turn evaluation of ai assistants? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35200–35278.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In *International Conference on Machine Learning*, pages 13481–13544. PMLR.

Lucie Flek. 2020. Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, Online. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Neil De La Fuente, Oscar Sainz, Iker García-Ferrero, and Eneko Agirre. 2025. GUIDEX: Guided synthetic data generation for zero-shot information extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24248–24262, Vienna, Austria. Association for Computational Linguistics.

Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. 2025. A survey on personalized Alignment—The missing piece for large language models in real-world applications. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5313–5333, Vienna, Austria. Association for Computational Linguistics.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, volume 2024, pages 34133–34156.

J Richard Hackman and Greg R Oldham. 1976. Motivation through the design of work: Test of a theory. *Organizational behavior and human performance*, 16(2):250–279.

Pengcheng Jiang, Jiacheng Lin, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2024. GenRES: Rethinking evaluation for generative relation extraction in the era of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2820–2837, Mexico City, Mexico. Association for Computational Linguistics.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of LLMs with mis-aligned responses. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11943–11974, Albuquerque, New Mexico. Association for Computational Linguistics.

Junseok Kim, Nakyeong Yang, and Kyomin Jung. 2025. Persona is a double-edged sword: Rethinking the impact of role-play prompts in zero-shot reasoning tasks. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 848–862, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Frederic Kirstein, Sonu Kumar, Terry Ruas, and Bela Gipp. 2025. Re-FRAME the meeting summarization SCOPE: Fact-based summarization and personalization via questions. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20087–20137, Suzhou, China. Association for Computational Linguistics.

Jinglin Liang, Yutao Qin, Shuangping Huang, Yunqing Hu, Xinwu Liu, Huiyuan Zhang, and Tianshui Chen. 2026. Knowledge-embedded graph representation learning for document-level relation extraction. *Expert Systems with Applications*, 295:128872.

Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew-Soon Ong. 2024. Large language models as evolutionary optimizers. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Meng Lu, Yuzhang Xie, Zhenyu Bi, Shuxiang Cao, and Xuan Wang. 2025. CROSSAGENTIE: Cross-type and cross-task multi-agent LLM collaboration for zero-shot information extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13953–13977, Vienna, Austria. Association for Computational Linguistics.

Pedro Henrique Luz de Araujo, Michael A. Hedderich, Ali Modarressi, Hinrich Schuetze, and Benjamin Roth. 2026. Persistent personas? role-playing, instruction following, and safety in extended interactions. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5329–5359, Rabat, Morocco. Association for Computational Linguistics.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.

Jisoo Mok, Ik-hwan Kim, Sangkwon Park, and Sungroh Yoon. 2025. Exploring the potential of LLMs as personalized assistants: Dataset, evaluation, and analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10212–10239, Vienna, Austria. Association for Computational Linguistics.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Sharon K Parker, Frederick P Morgeson, and Gary Johns. 2017. One hundred years of work design research: Looking back and looking forward. *Journal of applied psychology*, 102(3):403.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7957–7968.

Alireza Salemi, Julian Killingback, and Hamed Zamani. 2025. ExPerT: Effective and explainable evaluation of personalized long-form text generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17516–17532, Vienna, Austria. Association for Computational Linguistics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Alireza Salemi and Hamed Zamani. 2025. LaMP-QA: A benchmark for personalized long-form question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1159, Suzhou, China. Association for Computational Linguistics.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. 2025. Personax: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5764–5787.

Kai Shuang, Ji Zhou, Qiwei Wang, Bing Qian, Yunhao Wei, and Jinyu Guo. 2025. Utilizing contextual summarizing and reasoning for enhancing document-level event argument extraction. *Expert Systems with Applications*, 287:128075.

Hovhannes Tamoyan, Hendrik Schuff, and Iryna Gurevych. 2025. LLM roleplay: Simulating human-chatbot interaction. In *Proceedings of the Third Workshop on Social Influence in Conversations (SICon 2025)*, pages 1–26, Vienna, Austria. Association for Computational Linguistics.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. 2024. Two tales of persona in llms: A survey of role-playing and personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16612–16631.

Harsh Vishwakarma, Ankush Agarwal, Ojas Patil, Chaitanya Devaguptapu, and Mahesh Chandran. 2025.

Can LLMs help you at work? a sandbox for evaluating LLM agents in enterprise environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9167–9201, Suzhou, China. Association for Computational Linguistics.

Haochen Wang, Yi Wu, Daryl Chang, Li Wei, and Lukasz Heldt. 2026. Self-evolving recommendation system: End-to-end autonomous model optimization with llm agents. *arXiv preprint arXiv:2602.10226*.

Noah Wang, Z.y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. 2024. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Amy Wrzesniewski and Jane E Dutton. 2001. Crafting a job: Revisioning employees as active crafters of their work. *Academy of management review*, 26(2):179–201.

Tong Xiao, Jiayu Liu, Zhenya Huang, Jinze Wu, Jing Sha, Shijin Wang, and Enhong Chen. 2024. Learning to solve geometry problems via simulating human dual-reasoning process. *arXiv preprint arXiv:2405.06232*.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. 2022. GPS: Genetic prompt search for efficient few-shot learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8162–8171, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized generation in large model era: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24607–24649, Vienna, Austria. Association for Computational Linguistics.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: System demonstrations)*, pages 211–220.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. 2024. Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in neural information processing systems*, 37:43571–43608.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zheyuan Zhang, Lin Ge, Hongjiang Li, Weicheng Zhu, Chuxu Zhang, and Yanfang Ye. 2026. MAPRO: Recasting multi-agent prompt optimization as maximum a posteriori inference. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 4458–4480, Rabat, Morocco. Association for Computational Linguistics.

Guang Zhao, Byung-Jun Yoon, Gilchan Park, Shantenu Jha, Shinjae Yoo, and Xiaoning Qian. 2025a. Pareto prompt optimization. In *International Conference on Learning Representations*, volume 2025, pages 6770–6785.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. 2025b. PersonaLens: A benchmark for personalization evaluation in conversational AI assistants. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18023–18055, Vienna, Austria. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. 2025. Craft your dataset: Task-specific synthetic dataset generation through corpus retrieval and augmentation. *Transactions of the Association for Computational Linguistics*, 13:1693–1721.

A Persona Generation Prompts and Examples

Persona expansion prompt

```
You are a senior HR analyst. Given an O*NET role descriptor below, expand it into a concrete enterprise persona JSON with the following keys: role, seniority, primary_objectives (list), document_distribution (list of {type, share}), and preferences (3-6 declarative statements about which extracted artifacts the persona considers useful or noisy). Keep preferences actionable and falsifiable; avoid generic statements.

O*NET descriptor:
{onet_role_json}

Return one JSON object only.
```

Figure 4: Prompt used in Stage 2 to expand each O*NET role into a persona.

B Dataset Details

Persona pool composition. The final 292 personas distribute across SOC major groups roughly as: Management 18%, Business/Financial 21%, Computer/Math 22%, Engineering 14%, Sciences 10%, Legal 5%, Education 6%, Arts/Media 4%.

Splits. Splits are seeded by `persona_order_60_20_20` (deterministic). Train, dev, eval sets are used for inner-loop optimization, outer-loop evolution / early stopping, and final reporting respectively.

C Hyperparameters

Parameter	Value
Max optimization iterations T	30
Max inner-loop steps per persona	8
Success buffer size K	20
Meta-evolution frequency	every 5 iterations
Temperature	0.0
Max generation tokens	2048

Table 9: Hyperparameter configuration.

D Prompt Templates

Initial extraction prompt (abridged)

```
You are an expert analyst tasked with extracting the current user's project names and tasks. Read the provided enterprise artifacts and output exactly one valid JSON object that matches the schema below. Output JSON only.

## Objectives
- Identify projects the user actively works on and extract relevant tasks.
- Preserve exact project names from the artifacts.
- Rank projects by relevance based on ownership, urgency, recency, and leadership signals.

## Citing Sources
- Cite evidence using {cursor}†L{line_start}(-L{line_end})?.
- Include at least one citation for every project and task.

## Project Identity Gate
Tier-A: user involvement plus stable identity in ≥2 artifacts within 30 days.
Tier-B: one strong anchor authored/owned by the user within 14 days plus one weak corroborator.

## JSON Output Contract
- Return exactly one JSON object; double-quoted keys/values; no trailing commas; no markdown.
```

Figure 5: Abridged initial extraction prompt.

AI-User system prompt

```
You are role-playing the following person. Be accurate and self-consistent rather than polite. If something does not match your real experience, reject or correct it. Return exactly ONE valid JSON object.

Schema:
{ "project_feedback": [{"name", "decision":"keep|delete|revise", "reason", "suggestion"}],
"task_feedback": [{"title", "decision", "reason", "suggestion"}],
"overall_comment": "string" }
```

Figure 6: AI-User feedback prompt.

Inner-loop edit-plan prompt (excerpt)

```
You are a Prompt Optimization Specialist.
Produce a minimal Patch for the structured
prompt below based on AI-User feedback.
Pinpoint the exact (section, point_id)
responsible for the error and emit an edit plan
with action ∈ {replace_content, patch_logic,
add_constraint}. Return JSON only.
```

Figure 7: Inner-loop edit-plan prompt.

Outer-loop meta-evolution prompt

```
Current Meta-Prompt: {current_meta_prompt}
The following optimization steps were
SUCCESSFUL: {success_trace_json}
Analyze why and rewrite the Meta-Prompt to be
more precise, particularly in handling point_id
and instruction logic. Return only the new
Meta-Prompt text.
```

Figure 8: Outer-loop meta-evolution prompt.

E Cross-Model Robustness

To rule out the risk of self-preferential drift (Liu et al., 2023; Fu et al., 2024; Dou et al., 2025), we re-run the held-out evaluation using a non-GPT model as the AI User while keeping the extractor as GPT-5.1. With Claude-3.7 as judge, our SR is 0.728 (vs. 0.746 with GPT judge); with Qwen-2.5-72B, 0.713. Method ranking (ours $>$ ProTeGi $>$ Bandit-UCB $>$ OPRO $>$ ...) is preserved in both cases, indicating that the gains are not an artifact of judge–extractor coupling.

F Qualitative Error Analysis

We sampled 30 failed personas (deployed loss above τ after 20 iterations) and labelled their dominant failure mode. Distribution: preference conflicts 43%, domain-knowledge gaps 31%, sparse feedback 26%. Representative cases:

- *Preference conflict.* A "Senior PM" persona simultaneously prefers "brief actionable bullets" and "rich contextual notes," producing oscillating edits.
- *Domain knowledge.* A "Patent Counsel" persona expects extraction of priority dates and claim numbers; the base LLM lacks the schema to surface them reliably.
- *Sparse feedback.* A "Site Reliability Engineer" persona's documents contain mostly noise; the AI User issues few labels per step, slowing inner-loop convergence.

G Human Evaluation Details

Persona realism (5 evaluators). Evaluators were CS/Linguistics PhD students unaffiliated with this work. They rated 100 mixed samples (50 synthetic, 50 real LinkedIn-derived profiles) on Role Consistency, Technical Depth, Format Realism, and Language Naturalness using a 5-point Likert scale, double-blind. Cohen's $\kappa = 0.71$ (pairwise average), Fleiss' $\kappa = 0.68$.

Pairwise preference (20 evaluators). Five professionals each from engineering, PM, finance/legal, and editorial/research domains. Each evaluator saw 10 paired (baseline-prompt, ours-adapted-prompt) extractions for their own role, randomized in order, and selected "A wins / B wins / Tie." 200 judgments total, win rate 71.0%, $p < 10^{-5}$ (binomial test against 50% null).

Feedback authenticity (5 evaluators). 100 (persona, document, AI-feedback) tuples; classified as Suspected-AI / Uncertain / Human-like.

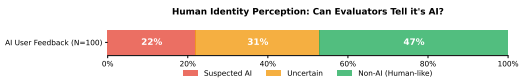


Figure 9: Distribution of human perception in the AI-User authenticity test ($N = 100$).

H Input/Output Examples

Input artifacts (excerpt)

```
=== ARTIFACT 1: {Type: email, Cursor: CUR00001} ===
L1 [2025-06-14T08:18] From: Ariana Brooks
<abrooks@techwave.com>
L2 To: Samuel Levin <slevin@techwave.com>
L4 Subject: Re: Model Registry Permissions
L7 Hi Sam, updated privileges for 'mlops-uat'. If you try again, should be
set now.

=== ARTIFACT 2: {Type: chat} ===
L1 [09:02] [Priya Desai] morning team
L3 [09:03] [Samuel Levin] deployed test script for S3 checks
L5 [09:03] [Max Tan] artifact storage full again?
```

Figure 10: Example synthetic enterprise artifacts.

```
{
  "projects": [{
    "name": "string",
    "project_type":
    ↪  "Project|Area|Initiative|Topic|AdminOps",
    "status": "Active|Blocked|Done|Paused|Unknown",
    "owners": ["person"],
    "tasks": [{
      "title": "string",
      "status": "Todo|InProgress|Blocked|Done|Unknown",
      "assignees": ["person_id"],
      "due_date": "YYYY-MM-DD|null",
      "priority": "P0|P1|P2|P3|Unknown",
      "evidence": ["Lx-Ly"],
      "confidence": 0.0
    }],
    "evidence": ["Lx-Ly"],
    "confidence": 0.0
  }]
}
```

Figure 11: Output JSON schema.

I Additional Evaluation

Public-document pilot. We additionally evaluate on public documents from CUAD[1] and SciREX[2], using 20 documents and five role-conditioned extraction objectives per dataset. Documents are kept unchanged; the personalized objectives are authored by us. Table 10 compares the initial prompt with the adapted prompt under our success criterion. Across 200 document–objective pairs, SR increases from 32.5% to 57.5%. This is a pilot-scale test on external documents, not a full evaluation under the original datasets' annotation schemes; its small size and author-designed objectives limit the conclusions about generalization.

Dataset	Initial	Ours	Δ
CUAD	35.0	55.0	20.0
SciREX	30.0	60.0	30.0
Aggregate	32.5	57.5	25.0

Table 10: Public-document pilot: SR (%) over 20 documents $\times$ 5 objectives per dataset. Δ is in percentage points.

Shared objective without personalization. To examine a setting in which all users want the same extraction, we replace the personas with a single generic analyst while holding the documents, extractor, and feedback loop fixed. Table 11 shows that the initial prompt already achieves 88.0% SR and meta-evolution adds 3.5 percentage points. The smaller gain is consistent with the framework's intended use for heterogeneous preferences; it does not establish a cost advantage when a common prompt already meets users' needs.

Method	SR (%)	ML
Initial Prompt	88.0	0.220
Self-Frozen	90.0	0.185
Self-Meta-Evolve	91.5	0.176

Table 11: Additional evaluation with a shared extraction objective.

One-shot editing and scope of comparison. The additional one-shot editing condition gives the model the initial structured prompt and AI-User feedback for the first document, and applies exactly one edit plan without iteration. This condition measures automated one-shot revision; it is not a comparison with an expert human prompt engineer. Establishing the benefit over capable human editing remains an open evaluation question. Simple requirements, such as extracting only items with an explicit owner and deadline, may be specified directly in one edit. Our motivation is to reuse editing experience across users with more complex or context-dependent preferences.

[1] https://www.atticusprojectai.org/cuad
[2] https://github.com/allenai/SciREX