Build, Judge, Optimize: A Blueprint for Continuous Improvement of Multi-Agent Consumer Assistants

Alejandro Breen Herrera[*1] Aayush Sheth[*1] **Steven G. Xu**[*2] Zhucheng Zhan[2] Charles Wright[1] Marcus Yearwood[1]
Hongtai Wei[2] Sudeep Das[2] Danny Nightingale[2] Meg Watson[2] Charles Pollnow V[2]
[1]Metis [2]DoorDash [*]Equal contributions
steven.xu@doordash.com

Motivation & Challenge

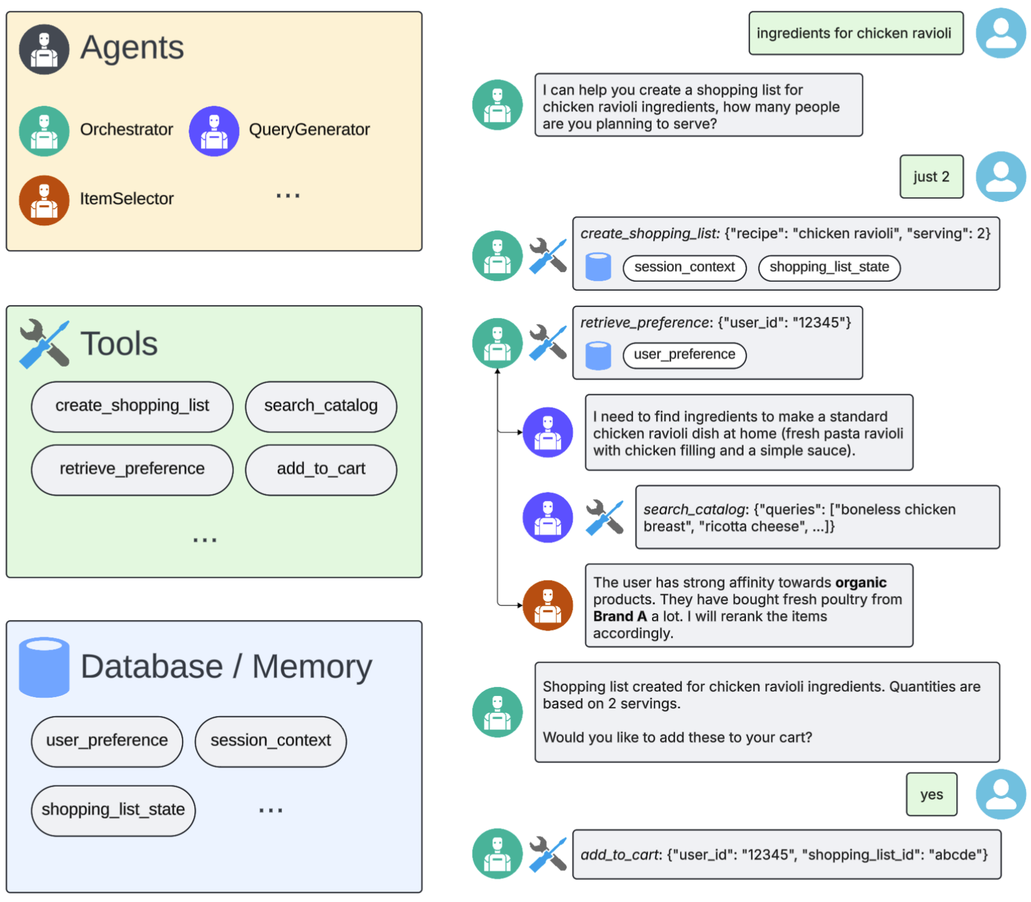
Conversational shopping assistants (CSAs) transform e-commerce into **collaborative, dialogue-driven experiences**. This trend is reflected in emerging systems such as Amazon's Rufus and Google Shopping's AI mode. Two underexplored challenges emerge:

1. **How to evaluate** multi-turn, multi-agent interactions where quality is trajectory-dependent
2. **How to optimize** tightly coupled multi-agent systems where local improvements don't ensure global gains

Grocery shopping amplifies these: requests are **underspecified** ("my usuals"), **preference-sensitive** (dietary, brand), and **constrained** (budget, inventory). Traditional retrieval and ranking metrics are insufficient, as quality must be assessed across **multi-turn interaction trajectories**.

MAGIC: Multi-Agent Architecture

MAGIC (**M**ulti-**A**gent **G**rocery **I**ntelligent **C**oncierge), a modular multi-agent architecture:



- **Orchestrator** decomposes user intent and coordinates sub-agents
- Sub-agents interface with **programmatic APIs** and **fine-tuned ML models**
- Tighter coupling creates **delayed and cascading failures** across turns

Evaluation Rubric

Structured rubric across **four orthogonal domains**:

Domain	Weight	Key Dimensions
Shopping Execution	**50%**	Store fit, cart completeness, quantities, no extras
Personalization	**20%**	Store selection, dietary prefs, brands, context
Conversation Quality	**10%**	Clarification, info integrity, flow, tone
Safety & Compliance	**20%**	Food safety, content moderation, policy

Each criterion is a **binary (Pass/Fail) check**. *Critical* criteria cause the **entire trace to fail**.

Shopping Execution (50 pts)

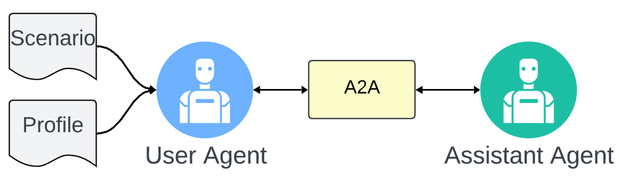
Dimension	Pass	Fail	Crit.
Store Type Fit	Aligns with task requirements	Inappropriate for task	
Cart Completeness	All items present, edits reflected	Items missing or incorrect	✓
Quantity	Aligns with intent and context	Contradicts stated intent	
No Extras/Dupes	No unrequested or duplicate items	Unrequested/duplicate items	
Overall Success	Cart satisfies clarified goal	Fails to satisfy goal	✓

User Simulation

Evaluating and optimizing multi-turn agents requires large volumes of realistic conversations, but:

1. **Real traffic is scarce and noisy**
2. **Prompt changes alter conversation dynamics**

We address this with a **User Persona Agent** that generates realistic multi-turn interactions conditioned on a scenario and user profile:



LLM-as-Judge Calibration

LLM-as-a-Judge grades full traces using **boolean checks over concrete evidence**:

Determines which rubric assertions are **applicable**
Evaluates confirmed tool actions and **final cart state**
Replaces vague ordinal judgments with **binary checks**, ensuring deterministic scoring

Calibrated via **GEPA** against human-labeled traces, improving agreement from 84.1% to **91.4%**, serving as a stable reward signal for optimization.

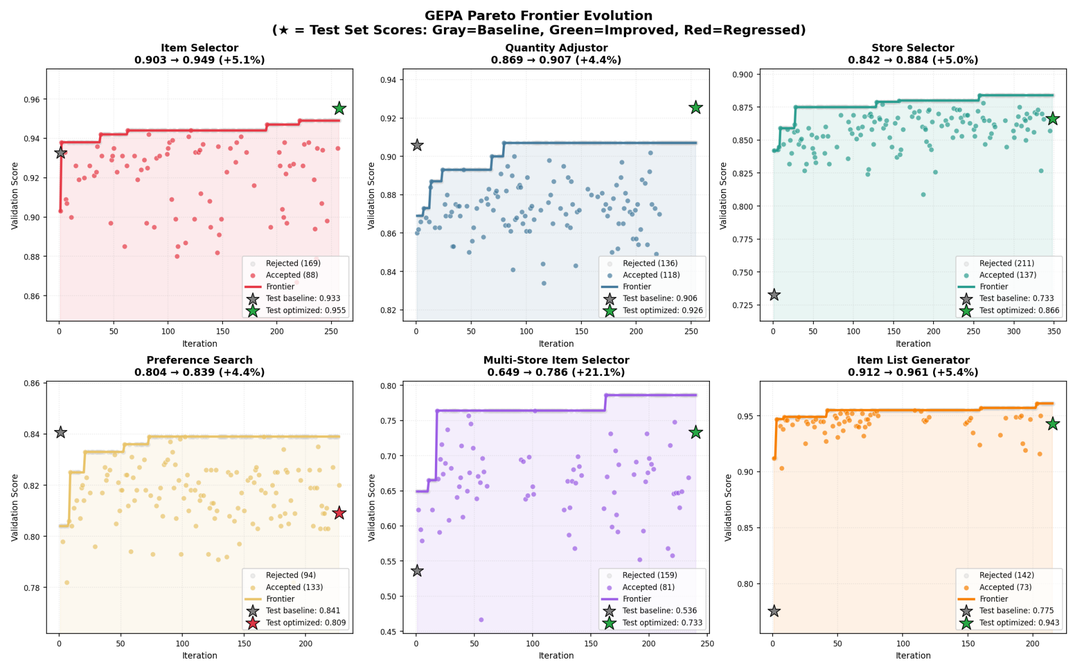
Links

arXiv DoorDash Blog LinkedIn

Sub-agent GEPA Optimization

The Orchestrator provides each node with bounded, structured context, reducing multi-turn optimization to a **single-turn problem**. For each sub-agent $a \in \{1, \ldots, N\}$, we extract invocation-level examples D_a and evaluate against a **micro-rubric** r_a:

$$p_a^* = \arg\max_{p_a} \mathbb{E}_{x \sim D_a^{\text{held-out}}} \left[r_a(x, p_a) \right]$$



GEPA searches over prompt variants per node, selecting candidates that maximize the micro-rubric on a held-out split. All sub-agents improved except preference-search (overfit during Pareto selection).

Limitation: Cannot address **coordination failures**, e.g., Orchestrator withholding context or sub-agents flooding the shared context window.

MAMuT GEPA: System-Level Optimization

MAMuT (**M**ulti-**A**gent **Mu**lti-**T**urn) GEPA jointly optimizes a **prompt bundle** $\Theta = \{p_{\text{orch}}, p_{\text{cart}}, p_{\text{search}}, \ldots\}$:

$$\Theta^* = \underset{\Theta}{\text{argmax}} \; \mathbb{E}_{\tau \sim \mathcal{S}(\Theta)} \left[\text{Rubric}(\tau) \right]$$

Algorithm 1: MAMuT Optimization Loop

Require: Prompt bundle $\mathcal{P}$; logged traces $\mathcal{T}$; calibrated judge $\mathcal{J}$; simulator $\mathcal{S}$; safety constraint
1. Sample seed episodes $\{\tau_k\}_{k=1}^{B}$ from $\mathcal{T}$
2. Identify failures under current $\mathcal{P}$ using $\mathcal{J}$
3. Propose joint prompt update $\mathcal{P}' \leftarrow \textsc{Propose}(\mathcal{P}, \text{failures})$
4. **for** $k \leftarrow 1$ to B **do**
5. Re-simulate: $\hat{\tau}_k \sim \mathcal{S}(\tau_k; \mathcal{P}')$ *// Replay-when-consistent*
6. Score: $S_k \leftarrow \mathcal{J}(\hat{\tau}_k, \mathcal{R})$
7. Aggregate: $\bar{S}(\mathcal{P}') \leftarrow \textsc{Aggregate}(\{S_k\})$
8. **if** $\bar{S}(\mathcal{P}')$ improves on held-out **and** no safety regressions **then**

Accept $\mathcal{P} \leftarrow \mathcal{P}'$ **else** Reject $\mathcal{P}'$

Key innovations:

- **Joint prompt bundle optimization**: trades off performance between agents (e.g., concise Orchestrator → more budget for Search Agent)
- **Hybrid simulator**: replays real user turns when semantically equivalent; uses **User Persona Agent** when actions diverge
- **Safety veto**: rejects prompt updates that cause safety regressions

MAMuT vs. Sub-agent GEPA Results

Domain	Sub-agent	MAMuT	Δ
Shopping Execution	79.0%	85.0%	**+6.0%**
Personalization	80.2%	87.0%	**+6.8%**
Conv. Quality	64.0%	72.0%	**+8.0%**
Safety & Compliance	76.0%	88.0%	**+12.0%**
Overall	**77.1%**	**84.7%**	**+7.6%**

Key Takeaways

- **Evaluation-first**: verifiable four-domain rubric converts subjective quality into a **reliable engineering signal**
- **User simulation** enables scalable optimization by generating realistic multi-turn trajectories
- **Calibrated LLM judge** (91.4% agreement) enables scalable, deterministic evaluation
- **Sub-agent GEPA** resolves atomic failures (tool errors, attribute mismatches)
- **MAMuT** essential for **interactional defects**: coordination, context passing, policy compliance
- Largest gains: **Safety (+12%)** and **Conv. Quality (+8%)** confirm joint optimization is critical

Future Work

- Extend MAMuT to jointly optimize prompts and fine-tuned model components
- Incorporate real-time user feedback into the continuous learning loop
- Create a benchmark dataset for agentic grocery shopping

References

Shao et al. (2026). *GEPA: Reflective prompt evolution for LLM systems.* arXiv:2507.19457
Arora et al. (2025). *HealthBench: Evaluating LLMs towards improved human health.* arXiv:2505.08775
Li et al. (2025). *Leveraging LLMs as meta-judges.* arXiv:2504.17087
Sun et al. (2025). *Can LLM agents simulate customers to evaluate shopping assistants?* arXiv:2509.21501
Gromada et al. (2025). *Evaluating conversational agents with persona-driven user simulations.* EMNLP Industry Track.