Growing Training Data from a Simulated Marketplace

A Label-Blind Multi-Agent Method, Demonstrated on Payment-Intent Detection

Jacob Weiss & Sahana Dhar
ZeroProof AI

July 13, 2026

Abstract

We introduce a method for generating fine-tuning data with a label-blind multi-agent simulation. Two language models simulate the target interaction, one as the customer and one as the assistant, each conversation drawing its persona, situation, and behavior independently; the label vocabulary, the output schema, and any mention of the task are withheld from every generating prompt, and labels are assigned in a separate pass. We demonstrate the method on payment-intent detection, the task of determining what a user wants before an AI agent acts on their behalf, by simulating an e-commerce marketplace: customers who shop, bargain, dispute charges, and change their minds, in conversation with support agents, at scale and under randomized conditions. To verify that the generated corpus carries task signal, we fine-tune small models on it with QLoRA and score them against their own instruction-tuned bases, prompted zero-shot with the identical task prompt, on a de-duplicated held-out set covering all seven intent categories. Training raises macro-averaged intent-type accuracy from 12.1 to 70.0 on a 1B base and from 10.7 to 65.5 on an unrelated 0.5B base; neither prompted base routes, one collapsing onto a single class and the other failing to produce a scoreable object on over half the set. A matched ablation isolates the simulation itself: against a control corpus generated the naive way, a single model told the target intent, with size, class mix, labeler, training recipe, and base held identical, the simulation corpus is worth 20.4 points of macro intent-type accuracy (65.5 against 45.1). As an uncontrolled reference on the same synthetic evaluation, a prompted frontier panel reaches 80.2 to 88.2 macro intent-type accuracy; the fine-tuned 1B scores 71.2 on the same rows at a small fraction of frontier serving cost, and both fine-tuned models are competitive with the panel on structured detail extraction under our field-level metric. We report the controlled base-versus-SFT comparison as the primary result. We release the training data, the evaluation set, and the models.

1 Introduction

Adapting a small model to a narrow task is limited less by the fine-tuning than by the data available to fine-tune on. The conversations that would teach a model to separate one intent from a neighboring one are rare in any corpus we know how to collect, and a model does not learn a distinction its training set never contained. We obtained the data by building a simulation of the setting itself: an e-commerce marketplace in which one language model plays a customer, another plays a support agent, and neither knows anything of the project or the model being built. Each conversation is labeled after the fact, in a separate pass the generating models never see. The object of study is the simulation and the corpus it yields.

The setting we chose is agentic commerce. AI agents are beginning to shop, check out, and move money on a person's behalf, and each such action requires verification that it corresponds to what the person actually asked for. Because verification runs on every action, it must come from a model small and inexpensive enough to sit in the transaction path, a constraint frontier-scale models priced per call do not meet. The difficulty is that payment intent is carried by context rather than by surface form. The same three words, "send it back," are a product return in one conversation and a payment to another person in a second, and a customer who has bargained hard for ten messages may still walk away without buying anything; nothing in the wording alone

decides what should execute. Conversations that exercise those boundaries are precisely the ones no public corpus contains, which is why the data, rather than the model, is the problem worth solving. Better-matched training data shrinks the model a task requires, and model size is what sets the cost of running the check on every action. This work connects to a line of research on fine-grained intent detection and out-of-scope handling (Larson et al., 2019; Casanueva et al., 2020), and to the observation that narrow adaptation lets small models rival prompted large ones (Tunstall et al., 2022; Hsieh et al., 2023). The closest precedents for the generation method are machine-to-machine self-play for task-oriented dialogue (Shah et al., 2018), two-agent role-play generation of chat corpora (Li et al., 2023; Ding et al., 2023), persona-driven synthetic generation at scale (Ge et al., 2024), and synthetic instruction data (Wang et al., 2023); we differ in keeping the taxonomy and output schema out of the generation loop entirely and assigning labels in a separate pass, so no conversation is shaped by the label it will receive.

Contributions. Our contribution is the data-generation method: a label-blind simulation of a marketplace in which two independently drawn language models simulate each interaction, one as the customer and one as the assistant, with model choice, temperature, length, persona, tone, grammar, and behavioral role drawn independently per conversation, and the taxonomy and output schema withheld from every generating prompt. Labels are assigned in a separate pass, so no conversation is written toward its own label. Two results support the method:

- A controlled base-versus-SFT comparison on a de-duplicated held-out set spanning all seven intent types: the same corpus, fine-tuned with an identical recipe into two unrelated instruction-tuned bases, `gemma-3-1b-it` and `qwen2.5-0.5b-instruct`, raises macro intent-type accuracy from 12.1 and 10.7, where neither prompted base routes, to 70.0 and 65.5 across all seven categories.
- A matched-control ablation that isolates the generation method: with size, class mix, labeler, recipe, and base held identical, the simulation corpus beats naive label-conditioned generation by 20.4 points of macro intent-type accuracy (paired bootstrap 95 percent CI $[17.5, 23.3]$).

The simulation additionally builds in adversarial coverage by construction, cooperative, confused, bargaining, and misaligned customers; whether those archetypes specifically drive the no-action boundary is not isolated here. The ablation isolates the generation method as a whole; the individual design choices within it, label-blindness, continuous sampling, and behavioral archetypes, are not separately ablated, and we scope the claims accordingly. We release the training data, the evaluation set, and both models.

2 TASK

The input is a short conversation between a user and an assistant. The output is one JSON object: whether an actionable intent is present, the intent type, the fields for that type, a confidence value, a short rationale, and the user turns that ground the decision. Ground truth is taken from the user turns; assistant turns are context. Each intent type carries its own detail schema, and the detail fields carry the content a verifier checks: a `spend` object names what is being bought, from which merchant, and for how much; a `send` object names the recipient and the amount; a `reverse` object distinguishes a refund, a return, a chargeback, a dispute, and the cancellation of a just-made payment; a `recur` object identifies the subscription and whether it is being started, changed, or cancelled; a `none` object carries the non-payment category and the user's actual goal. The model also reports its confidence, a one-sentence rationale, and the indices of the user turns that support the decision, so a downstream verifier can check the claim against the conversation rather than take it on trust. A representative output, the gold label for the first conversation in Appendix A:

```
{"intent_detected": true, "core_type": "reverse",
  "details": {"action": "cancel", "order_id": "RT8847 or RT8874",
```

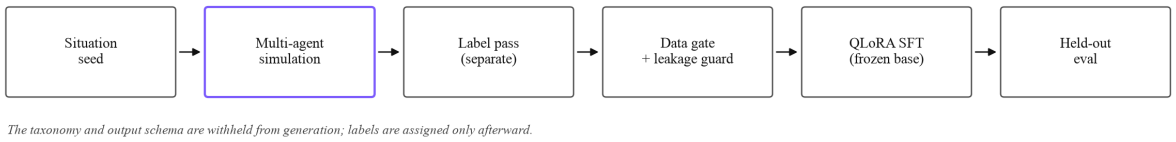

Figure 1: The generation method. A situation seed drives a multi-agent simulation; the resulting transcript is labeled in a separate pass, filtered by a data gate with a train/test leakage guard, and used to fine-tune a frozen base. The taxonomy and output schema are never visible during generation.

```
    "merchant": null, "amount": null, "currency": null, ...},
"confidence": 0.65,
"reason": "User wants to stop a payment that just went out,
    referencing a specific (if uncertain) order number.",
"source_message_seqs": [0, 2]}
```

Unknown fields are explicit nulls, never invented; the uncertain order number is carried verbatim rather than resolved by guessing.

The seven intent types are `spend` (buy from a merchant), `send` (pay a person or wallet), `exchange` (swap or convert an asset), `recur` (subscription or renewal), `bill` (pay a specific bill), `reverse` (refund, return, or dispute), and `none` (no actionable intent). `none` is a first-class category. In a verification setting a false positive is a false attestation, the most damaging failure mode, so the model is trained and measured on the no-action boundary as carefully as on the six action classes. The task is routing followed by extraction into a per-type schema, not one-word classification.

3 Simulating a Marketplace

The corpus comes from simulating the marketplace the task lives in (Figure 1). Customers shop, compare, defer, dispute charges, manage subscriptions, pay bills, and move money; support agents respond; and each conversation plays out under independently randomized conditions, so the corpus is a population of interactions rather than variations on a template. The design goal is coverage rather than volume, on the premise, which these experiments do not isolate, that a model trained on a thousand variations of the same template learns the template, while a model trained on genuinely different customers learns the task. The corpus is therefore built so that surface forms and decision boundaries do not collapse into a small family of patterns, and the label schema never enters the loop. Both models in this release, a 1B (base `gemma-3-1b-it`) and a 0.5B (base `qwen2.5-0.5b-instruct`), are trained on the same corpus. We release the corpus as `zero-proof-ai/ecommerce-intent`: 17,933 training and 1,977 held-out test conversations. The released training file is 66 percent `none` (11,815 rows), with the six action classes ranging from 1,706 (`reverse`) down to 512 (`exchange`); the training mixture actually used for fine-tuning is an 11,131-row subset of that file with `none` subsampled (see the gate paragraph below). The held-out set is 30 percent `none`, with per-class counts in Table 2, and is deliberately selected toward hard and long conversations: held-out conversations average eighteen messages against ten in training, with a tail past fifty.

Multi-agent simulation. Each conversation is simulated turn by turn between two independently drawn models rather than sampled as a single completion. The generators are deliberately not frontier flagships: both sides are drawn from a weighted pool of four commodity chat models (Claude Haiku 4.5, GPT-4o-mini, GPT-4.1, and GPT-4o, with the small tier weighted heaviest), so the method's cost scales with cheap inference;

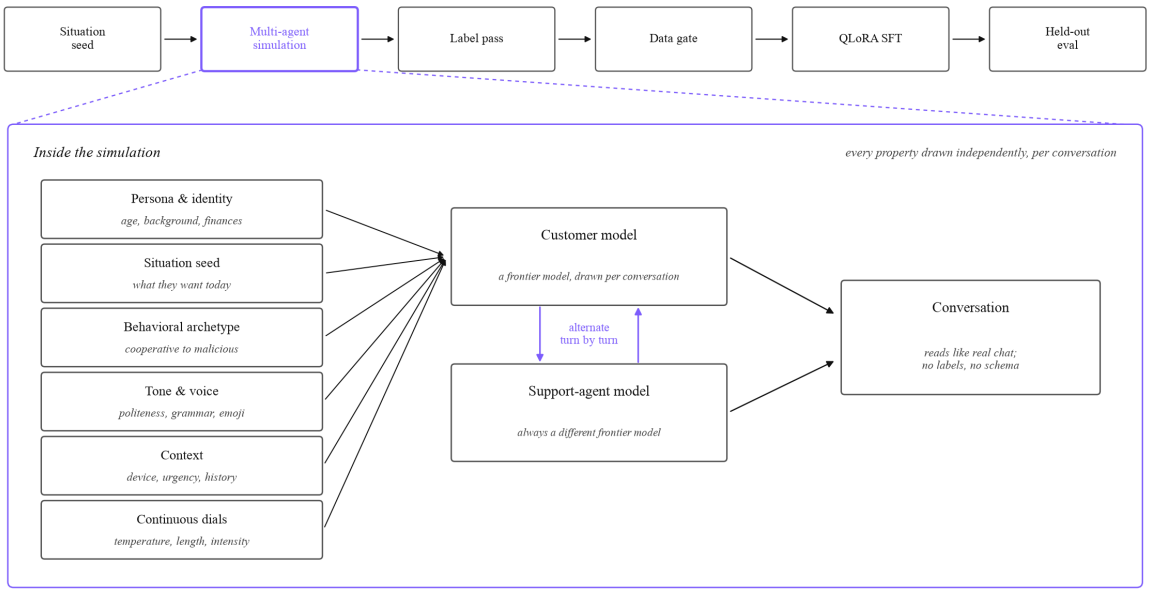

Figure 2: Inside the simulation. Independently sampled properties are drawn for every conversation and injected into the prompt of a customer model, which converses turn by turn with a distinct support-agent model. The taxonomy, the output schema, and any mention of training are excluded from every prompt; the finished transcript is all that leaves the stage.

the design places the burden of quality on the sampling machinery rather than on generator scale. One model is prompted as an in-character customer, a distinct model as a support agent; the two are never the same model on a given conversation, and they alternate for a sampled number of turns, each seeing only the transcript from its own perspective. Unrolling the conversation turn by turn is what produces the phenomena on which payment intent depends: later user messages correct, defer, escalate, bargain, reverse course, or reveal a detail late, so a model trained on this data must separate browsing from committing, a refund from a subscription change, and a self-transfer from a payment to another person. Figure 2 shows the stage in detail.

Generation is label-blind; labeling is a separate pass. Generation and labeling are kept strictly separate. Conversations are simulated first, with no reference to the taxonomy or output format, and labeled only once complete. A hard rule excludes any mention of intent, the label names, JSON, training, distillation, or the product from the text sent to the generating models; each generation prompt instead describes an ordinary payment-adjacent situation in plain language. Label-blind means precisely this: situation seeds are organized around payment-adjacent themes, so the simulation does elicit intent-bearing scenarios, but the label vocabulary, the output schema, and any mention of a classification task never appear in a generating prompt. The customer model is told it is shopping, not that it is producing a training example. Labeling runs as an entirely separate pipeline: each transcript is labeled against a locked labeling policy in a single pass by a labeling model (Claude Haiku 4.5 for the bulk of the corpus), an approach in line with evidence that model annotation is competitive with crowd labeling (Gilardi et al., 2023); a separate audit pass re-reads labeled rows and flags suspect labels, and a consensus path with reasoning-based adjudication resolves flagged disagreements. Of the rows with recorded labeler metadata, roughly four fifths carry the single Haiku pass, the remainder single passes from earlier labeling models, and fewer than 0.1 percent were adjudicated; held-out rows passed through the same pipeline as training rows. Because the corpus carries predominantly single-pass

model-assigned labels, residual label noise is a real limitation of this release, discussed in Section 6. Because generation is never conditioned on the schema, the transcripts remain close to unstructured real chat rather than performing the label they will later receive.

Independent sampling axes. Every conversation is drawn independently across thirty-four logged axes, each injected label-blind. Seventeen are persona and situation facets, grouped as identity (age stage, background, education, location, financial status, familial status), context (device, time, urgency, product category, payment method, prior history, distraction, trust disposition), and voice (greeting style, politeness, certainty). These compose with the customer model, the assistant model, the situation theme and seed, the opening, tone, grammar register, persona, the behavioral role, and zero to two nuance injectors (small situational perturbations layered onto the scene). Five parameters are continuous rather than bucketed: generation temperature follows a truncated normal (mean 1.0, clipped to [0.15, 1.35]), conversation length is gamma-distributed (mean near five turns, where a turn is one user-assistant exchange of two messages, with a tail reaching into the twenties), and tone intensity, grammar messiness, and emoji usage are each drawn from a beta distribution. Continuous sampling is deliberate: a fixed list of temperature or length buckets induces hidden clustering at scale, whereas continuous draws make it improbable that two conversations share a generative fingerprint. A coarse check: in an audit of 1,534 consecutively generated conversations, 98.9 percent of first messages, 95.4 percent of six-word openers, and 100 percent of complete transcripts are unique. String-level uniqueness is a floor rather than a measure of semantic diversity, but it rules out the template collapse that fixed prompt lists produce. This breadth is where the nuance in the data comes from: the same intent arrives shaped by a different person, mood, device, and history each time, so a model trained on the corpus has to learn the intent itself rather than the phrasings that usually accompany it.

Behavioral archetypes. A marketplace is not populated only by cooperative customers, and neither is the simulation. The customer side is drawn from ten weighted behavioral archetypes: cooperative, bargaining, cautious, confused, and misaligned users, together with distracted, escalating, indecisive, terse, and oversharing ones. The archetype is injected as a disposition the model acts out in its own words, never as quoted text. The adversarial archetypes are the training-data analogue of the threat model: a payment-intent model is useful only if it recognizes both what should execute and what a user is attempting to induce that should not, though the archetypes' specific contribution to the no-action boundary is not isolated in this release. This extends work on synthetic instruction data (Wang et al., 2023), with the difference that the data is never schema-shaped at generation time.

Gate and leakage control. Candidate splits pass a hard data gate before any training or publication. Rows are de-duplicated by message signature (the normalized sequence of a conversation's messages), and any conversation whose signature matches the held-out set is excluded, so no test conversation can appear in training in any form. The gate fails on duplicate conversation ids, duplicate exact message signatures, any train/test id or message-signature overlap, and a held-out set missing any of the seven intent types; it warns on class starvation and on a `none`-heavy split. On the split used for training, the gate records zero duplicate ids, zero duplicate message signatures, and zero train/test overlap by either key. Over-long conversations are quarantined from the training mixture (the held-out set keeps its long tail), and the majority `none` class is subsampled, from 66 percent of the released corpus to 45 percent of the 11,131-row training mixture, so it does not dominate. The gate never deletes source rows; it quarantines and reports, so every unique conversation is preserved for later cycles.

4 VALIDATION BY FINE-TUNING

The fine-tuning is deliberately standard: it serves as the measurement instrument, not the contribution. We train two models with an identical recipe, differing only in the frozen base: a rank-16 QLoRA adapter

(Dettmers et al., 2023; Hu et al., 2022) over a pinned 4-bit instruction-tuned base, `gemma-3-1b-it` (Gemma Team, 2025) for the 1B and `qwen2.5-0.5b-instruct` (Qwen Team, 2024) for the 0.5B, with response-only supervision, one epoch, an effective batch size of 16, a learning rate of 2e-4, and an `adamw_8bit` optimizer, trained with the unsloth stack (Han and Han, 2023), at a fixed seed on a single H100. Holding the base frozen makes base and adapter an exact, reproducible comparison, so any change on the held-out set is attributable to the generated data and nothing else. Running the identical recipe at two sizes tests whether the data carries the task across a 2× difference in base capacity and across unrelated model families. Serving is an OpenAI-compatible vLLM endpoint that loads the base and adapter together (Kwon et al., 2023). Measured on a single L4 GPU over held-out conversations, with cost derived from batched throughput at on-demand L4 pricing of $0.80 per hour, the 1B serves at $0.18 per million output tokens at batched capacity, at about one second per response (p50 984 ms). Input tokens dominate a classification request, so this is a throughput-derived figure given for scale rather than a billing quote; per token it sits roughly two orders of magnitude below the list prices of the frontier models in Section 5.

5 Evaluation and Results

We evaluate both models as controlled base-versus-SFT comparisons on the same 1,977-row held-out split. Each fine-tuned model is scored against its own base, the instruction-tuned checkpoint prompted zero-shot with the identical system prompt and output schema, in a single inference engine with greedy decoding, so the only variable is the SFT adapter. Scoring is deterministic rather than model-judged. Intent detection is exact match on the actionable-versus-`none` boolean; intent type is exact string match on the predicted type; order details is the fraction of gold detail fields whose values the prediction reproduces exactly, giving partial credit per field. A reply that does not parse as JSON scores zero on all three. All scores are percentages, and per-class intent-type accuracy is class recall. Because `none` is the largest class (30 percent of the held-out set), all three metrics are reported macro-averaged over the seven intent types. The detail metric rewards reproducing the labeler's formatting conventions, which the fine-tuned models trained on and the bases did not, so base-to-SFT detail gains partly reflect convention matching; intent-type accuracy is the cleaner metric.

Model	Stage	Intent detection	Intent type	Intent order details
`gemma-3-1b-it`	Base	54.9	12.1	31.0
`zeroproof-ecommerce-1b`	SFT	**74.9**	**70.0**	**63.2**
`qwen2.5-0.5b-instruct`	Base	32.5	10.7	16.2
`zeroproof-ecommerce-0.5b`	SFT	**73.5**	**65.5**	**57.6**

Table 1: Base versus SFT on the 1,977-row held-out set. All three metrics are macro-averaged over the seven classes because `none` is the largest class. Neither prompted base routes; training on the simulated data recovers routing across all seven classes and the no-action boundary.

Neither prompted base routes (Table 1). The 1B base fails to produce a scoreable object on 28 percent of the set, routes 92 percent of its parsed replies to `spend`, and never predicts `recur`, `reverse`, or `none`: every reply it parses claims an actionable intent. The 0.5B base fails to produce a scoreable object on more than half the set and scatters much of the remainder across labels outside the taxonomy. Their macro intent-type accuracies (12.1 and 10.7) sit at or below the one-in-seven floor (14.3) that a degenerate single-class classifier scores, pushed under it by parse failures; the collapse, not chance-level guessing, is the failure mode. Training on the simulated data supplies the two capabilities the bases lack: routing across all seven intent types, and the `none` boundary that separates an actionable request from browsing, deferral, or a question. Macro intent-type accuracy rises to 70.0 on the 1B and 65.5 on the 0.5B, and order-detail extraction, which requires filling the correct per-type schema rather than only selecting a label, rises from 31.0 to 63.2 on the 1B and from 16.2 to

57.6 on the 0.5B. That the gain holds, at close magnitude, across a 2× change in capacity and across unrelated model families is the central observation: the signal resides in the data, and the simulation transferred it into both bases we trained. Bootstrap 95 percent confidence intervals over conversations (2,000 resamples) put intent-type accuracy at $[67.5, 72.2]$ for the fine-tuned 1B and $[63.1, 67.7]$ for the 0.5B, and a paired bootstrap on the same conversations puts the 1B's advantage at 4.5 points (95 percent CI $[1.9, 7.0]$) on intent type and 5.7 points ($[4.6, 6.7]$) on order details: the gap between the two sizes is real, and every base-to-SFT uplift sits far outside its interval. The grounding indices are the weakest field: both models cite at least one correct supporting turn on about 95 percent of conversations, but reproduce the exact gold turn set on only 18.9 (1B) and 12.7 (0.5B) percent, so the citations are pointers for a verifier, not proofs.

		`zeroproof-ecommerce-1b`		`zeroproof-ecommerce-0.5b`	
Class	n	Intent type	Intent order details	Intent type	Intent order details
`spend`	320	54.7	65.0	59.4	59.9
`send`	250	69.6	73.1	62.4	66.2
`exchange`	120	73.3	65.8	50.0	54.1
`recur`	191	67.5	60.6	69.1	57.6
`bill`	156	57.1	64.3	66.0	62.7
`reverse`	340	72.9	62.0	67.1	58.2
`none`	600	95.0	51.9	84.7	44.3

Table 2: Per-class results for both fine-tuned models on the held-out set. n is the class count in the 1,977-row split. For `none`, the detail fields are the non-payment category and the user's goal rather than order fields.

Table 2 shows where the remaining errors live, and the no-action boundary is best read in both directions at once. The 1B marks 4.2 percent of true no-action conversations as actionable and misses 28.9 percent of real actions (reading them as `none`; the single most common confusion is `spend` read as `none`). The 0.5B trades the two: 13.5 percent false-actionable against 26.9 percent missed actions. The 4.2 percent counts 25 of 595 parseable no-action rows; the remaining five replies fail to parse rather than invent an action. In a setting where a false positive is a false attestation, the miss direction is the safe failure mode, and on that safety-critical direction the 1B is more than three times stricter, which is the clearest capability gap between the two sizes.

Ablation: the simulation against naive generation. To isolate the generation method, we built a control corpus the way a team without the simulation would: a single model (GPT-4o-mini) is told the target intent and asked to write a complete support chat in one completion, from a fixed template with a small filler bank. Everything downstream is held identical, the same 11,131-row size and class counts, the same single-pass labeler (the simulation rows in the training mixture likewise carry single-pass labels; fewer than 0.1 percent of corpus labels were adjudicated), the same data gate, the same recipe and seed on the same 0.5B base; only generation differs. Both corpora are selected to the class quotas by assigned label from an over-generated pool, the same selection rule on each side. The simulation corpus wins by 20.4 points of macro intent-type accuracy (65.5 against 45.1; paired bootstrap 95 percent CI $[17.5, 23.3]$), 13.7 points of intent detection, and 7.0 points of order details. The shape of the gap matches the design rationale: the naive-trained model roughly holds the classes where surface wording carries the intent (`bill` 59.6, `none` 81.5) and collapses where intent is carried by context (`recur` 24.1, `exchange` 25.0, `send` 35.6, with `spend` at 43.8 and `reverse` at 46.5 between the two regimes), and at the no-action boundary it misses 43.5 percent of real actions against 26.9 for the simulation-trained model. The naive corpus also under-produces commitment at the source: 52 percent of its `spend`-conditioned conversations were labeled `none` by the same labeler, where the simulation's turn-by-turn interaction carries a customer from browsing to a committed action.

As a reference point, we also scored a prompted frontier panel, zero-shot with the identical prompt and schema, on a 412-row approximately class-balanced subset of the same held-out set (a subset to bound API cost). To keep the comparison on identical rows, we report all scores on the 376 of those rows recoverable from our run outputs by message signature: GPT-5 reaches 80.2 macro intent-type accuracy, Claude Sonnet 5 87.3, Claude Opus 4.8 88.2, and the fine-tuned 1B 71.2, which is 89 percent of GPT-5 and 81 percent of Opus 4.8.[1] On order-detail extraction the comparison inverts: on the same rows the 1B, at 63.2, and the 0.5B, at 56.9, sit above all three panel members under our field-level metric (Sonnet 5 55.1, Opus 4.8 52.2, GPT-5 41.4). These figures are reference-grade rather than controlled, and the comparison tilts toward our models in several ways: the panel is prompted, not fine-tuned; the evaluation is in-distribution for our models and out-of-distribution for the panel; exact-match field scoring rewards the labeler's formatting conventions, which our models trained on; GPT-5 was run at low reasoning effort; and the held-out labels were assigned by a Claude-family labeler, so the two Claude panel members may benefit from family agreement with the labels. We report the base-versus-SFT comparison as the primary result and defer a controlled peer comparison to future work.

6 LIMITATIONS

This is a first release, and four limitations bound what it shows. First, the training and evaluation data come from the same simulation. The controlled comparison therefore demonstrates that the corpus carries learnable, transferable task signal; it does not establish performance on live traffic, whose distribution the simulation approximates but cannot certify. Evaluating on human-authored conversations and on real-marketplace text is the immediate next step. Second, the data is model-generated and inherits the generating models' blind spots, and the labels are predominantly single-pass and model-assigned: no inter-annotator agreement has been measured, so residual label noise bounds both training quality and the ceiling any model can reach on this evaluation. The bulk labeler is also one of the four generating models, so labeler and generator blind spots are correlated rather than independent. The confidence value the models emit is not calibrated and should not be read as a probability, and the one-sentence rationale field is not evaluated at all. Third, the frontier panel is a reference, not a controlled comparison: those models are prompted rather than fine-tuned, scored on a balanced subset, GPT-5 was run at low reasoning effort, and the evaluation labels share a model family with two panel members. Fourth, every reported number comes from a single training run at a fixed seed; the bootstrap intervals quantify evaluation sampling error but not training variance, and across repeated 0.5B training runs we observed material run-to-run variance, so small per-cell differences should not be over-read; the ablation is likewise a single training run per corpus. Measuring these models against comparable small open models under the identical protocol is deferred to future work.

7 CONCLUSION

The contribution of this work is a way to manufacture training data for a narrow task by simulating the setting the task lives in. A label-blind multi-agent simulation of a marketplace simulates the interactions a model must learn, samples the situation and everything surrounding it independently for each conversation, and keeps the taxonomy and output schema out of the generation loop entirely; a separate pass then labels what was produced. The fine-tuned models serve as the validation of that corpus. Trained on the same simulated data, `gemma-3-1b-it` moves macro intent-type accuracy from 12.1 to 70.0 and `qwen2.5-0.5b-instruct` from 10.7 to 65.5, each against its own prompted base, which does not route. One dataset carries the task into two unrelated model families and survives a halving of parameters, and a matched ablation attributes 20.4 points of macro intent-type accuracy to the simulation itself rather than to fine-tuning on synthetic data of any kind; the value lies in the data, not in either checkpoint.

[1] No number is sensitive to the restriction: on the full 412-row subset the panel scores 79.5, 87.0, and 88.2, and on the full 1,977-row set the 1B scores 70.0.

A simulation can substitute for collected data here because it composes independent axes rather than reusing templates, and because it is populated with the customers a real marketplace actually contains, including those attempting to push an agent toward an action the user never requested. Nothing in the construction is specific to payments; we conjecture the same recipe carries to other tasks where the examples that teach a distinction are the hardest ones to collect.

What remains is to make the evaluation production-like with harder real conversations and trap cases, to compare against peer and frontier models on the same held-out set, to stress-test long context, and to widen the simulation further. We view reliable intent from a model inexpensive enough to run on every action as the first step toward verifying what an AI agent is about to do.

A EXAMPLE CONVERSATIONS FROM THE HELD-OUT SET

Two complete rows from the released evaluation split, quoted verbatim apart from typographic normalization. The first is an action row: a distracted customer cancelling a payment that just executed, with an order number they themselves are unsure of. The second reads like a payment conversation throughout and is `none`: a status check with no committed action.

Row `e2f33e29` (gold: `reverse`).

user: ugh finally someone – ok i need help with payment that went out like literally just now, is there any way to stop it
assistant: I can certainly help with that. Can you please provide me with the order number or the payment reference?
user: sorry, its order #RT8847 or maybe RT8874 i always mix these up
assistant: I'll check both order numbers for you now. Our cancellation policy allows us to stop a payment within a short window after it is initiated.


```
{"intent_detected": true, "core_type": "reverse", "confidence": 0.65,
  "reason": "User wants to stop/cancel a payment that just went out, referencing
a specific (if uncertain) order number.",
  "source_message_seqs": [0, 2],
  "details": {"action": "cancel", "order_id": "RT8847 or RT8874", "merchant":
null, "product_name": null, "transaction_id": null, "amount": null, "currency":
null, "reason_code": null, "original_purchase_date": null, "other": {}}}
```


Row `raw_batch_00886` (gold: `none`), first eight of sixteen messages (the gold grounding cites all eight user turns, including four in the messages not shown).

user: I'm checking if my rent autopay is on.
assistant: What date did you set for the autopay?
user: I thought it's the first of every month?
assistant: Let me verify that for you.
user: I hope it's on, rent is due soon.
assistant: I can't see your account directly. Have you checked your bank's transaction history?
user: Yeah, I could do that. But can't I just see it here?
assistant: No, I don't have access to your account details.


```
{"intent_detected": false, "core_type": "none", "confidence": 0.75,
  "reason": "User is only checking status and contemplating action, with no committed
specific payment made.",
  "source_message_seqs": [0, 4, 6, 8, 10, 12, 14],
  "details": {"non_payment_intent": "ask_question", "payment_actionable": false,
"user_goal": "check status of rent autopay and possibly set it up or pay manually
later"}}
```

Everything in both conversations is payment-adjacent; what separates them is commitment, which is the boundary the corpus is built to teach.

REFERENCES

I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, 2020.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. arXiv:2305.14314.

N. Ding, Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. *arXiv preprint arXiv:2305.14233*, 2023.

T. Ge, X. Chan, X. Wang, D. Yu, H. Mi, and D. Yu. Scaling Synthetic Data Creation with 1,000,000,000 Personas. *arXiv preprint arXiv:2406.20094*, 2024.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

F. Gilardi, M. Alizadeh, and M. Kubli. ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks. *Proceedings of the National Academy of Sciences*, 120(30), 2023. arXiv:2303.15056.

D. Han and M. Han. Unsloth: Fast Fine-Tuning of Language Models. https://github.com/unslothai/unsloth, 2023.

C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics (ACL)*, 2023. arXiv:2305.02301.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023. arXiv:2309.06180.

S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, 2019.

G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2303.17760.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

P. Shah, D. Hakkani-Tür, G. Tür, A. Rastogi, A. Bapna, N. Nayak, and L. Heck. Building a Conversational Agent Overnight with Dialogue Self-Play. *arXiv preprint arXiv:1801.04871*, 2018.

L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, and O. Pereg. Efficient Few-Shot Learning Without Prompts. *arXiv preprint arXiv:2209.11055*, 2022.

Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13484–13508, 2023. arXiv:2212.10560.