Structured Tactics with Metamorphic Regularization for Training-Free Context Optimization

Anonymous ACL submission

Abstract

In this work we investigate training-free context optimization techniques for low-data intent text classification using Large Language Models (LLMs) as classifiers. While LLMs can be used as classifiers, in environments of low data they struggle to differentiate semantically adjacent labels, and optimizing the context for inference time can result in context-space overfitting, where the focus shifts to lexical pattern instability rather than to stable label semantics. The objective of this study is to compare fixed-artifact prompt optimization with iterative experience context optimization in deterministic classification tasks under metamorphic robustness tests. The proposed method, LEDSET, splits the inference context into three layers. First a task-level meta-context, secondly semantic invariants, and lastly mutable tactics are produced. The first two layers are stable, and only the last layer is optimized through an additive local confusion-cell repair regime. We conduct experiments using five CLINC150 subsets and an ultrasonic NDT digital twin intent-routing dataset. The experiments demonstrate that the iterative experience-based optimizers perform consistently better than the fixed-artifact prompt optimizers. The ablation results indicate that focusing only on last layer optimizations may cause an overemphasis on local confusion-pair relationships. However, combining both second and last layer optimizations provides stress robustness benefits. Finally, SFT+LEDSET outperforms LoRA-SFT via improved exploration of the prompt space.

1 Introduction

Large Language Models (LLMs) have the capability to be deployed as user intent classifiers. However, there are challenges in low data available scenarios, such as distinguishing semantically adjacent labels, taking care of text noise, handling adversarial prompts, and rejecting Out-Of-Scope (OOS) input texts. Publicly accessible pretrained models lack the boundary knowledge required for the decisions in specific tasks. This causes weak separability with scarce data and ambiguous label boundaries (Zhang et al., 2021). This difficulty is reflected in realistic intent classification benchmarks such as CLINC150 (Larson et al., 2019) and BANKING77 (Casanueva et al., 2020).

Although SFT can boost separability and domain fit, it does not assure that the model has received an exact decision logic at hard label boundaries. Similarly, while synthetic supervision can increase the amount of training data regarding new topics, it could potentially train generator-specific lexical artifacts rather than generalizable task semantics. With low data availability in intent classification tasks, many errors do not arise from representation errors alone. They are the byproduct of decision control as there are no explicit and auditable decision rules for the label boundaries in semantically adjacent cases. Thus, a different domain adaptation strategy is motivated, improving inference-time context instead of only changing the model's parameters. Yet, optimizing context instead of parameters comes with a cost, context space overfitting. Optimized prompts can overfocus on local words without grounded and stable decision rules.

This study explores training-free adaptation for low data text classification. We divide training-free adaptations into two families of methods. The first family is fixed artifact prompt optimization methods which optimize a fixed prompt artifact that will be used during all future inferences, including at test time. The second family are the iterative experience optimization methods that iteratively optimize an experience representation until the experience representation is frozen to represent the model's final learned experience including Reflexion, Self-Refine and TF-GRPO.

In this work, we study a progression from fixed-artifact prompt optimization to rollout-based experience induction, error-driven experience optimiza-

tion, and finally the ADD-only regime of Layered Experience Discovery with Structured Edit Tactics (LEDSET). Our method breaks down the inference context into the mutable tactics and the invariant inferences for each edit. It then locally repairs the tactics via an additive approach while using a confusion cell gate that has a per-edit activation rule. The empirical results show that LEDSET performs best in terms of MACRO F1 performance when comparing all of the presented methods. Furthermore, LEDSET provides the strongest performance among the iterative experience optimizers, while SFT+LEDSET reaches comparable scores with fewer optimization tokens due to the lower training error rate of the finetuned base model. Also, stacking LEDSET atop a LoRA-SFT prediction base improves both nominal accuracy, while the proposer LLM remains not optimized. In this paper, we make these contributions below:

- **The Context Optimization Ladder:** Methodological progression for low data text classification using fixed artifact prompt optimization to rollout experience induction (TF-GRPO-C) to error driven experience optimization (EDEO) to structured experience optimization with automated invariant discovery and local atomic tactical repair, tested here in LEDSET's ADD-only iteration regime.

- **LEDSET:** Stable semantic invariants vs. changing tactical heuristics with automated discovery of stable layers and local additive (ADD) only tactical edits with per-edit confusion cell gating.

- **Combining training-free optimization techniques on top of SFT with an un-tuned proposer:** The LEDSET pipeline integrates well with LoRA SFT of the prediction base. However, the proposer LLM is not finetuned.

2 Related Work and Problem Setting

2.1 Related Work

Recent automatic prompt optimization techniques and frameworks form a large family, as surveyed in (Ramnath et al., 2025). Certain examples of various representative methods include discrete search of prompts via gradient-guided search with AutoPrompt (Shin et al., 2020), model-generated instruction search with APE (Zhou et al., 2023), natural language optimization with OPRO (Yang et al., 2024), edit-based search for prompts with GrIPS (Prasad et al., 2023), critique-driven textual gradient with APO / ProTeGi (Pryzant et al., 2023), tree-search prompt planning with PromptAgent (Wang et al., 2024) and metric-driven compilation in DSPy (Khattab et al., 2024), and DSPy's MIPROv2 (DSPy, 2026b), COPRO (DSPy, 2026a), and GEPA (Agrawal et al., 2026). A limitation with these optimization families is they create a monolithic representation that incorporates all aspects of the task description, label semantics, and decision-making heuristics. Consequently, an edit to improve performance regarding one failure mode may negatively impact performance related to other types of input.

Textual feedback and self-revision can enhance parameter-frozen models without weight updates as shown in Reflexion (Shinn et al., 2023) and Self-Refine (Madaan et al., 2023), intrinsic self-correction (Huang et al., 2024). TF-GRPO examines reinforcement learning style adaptation as the induction of textual experiential knowledge from groups of sampled rollouts (Cai et al., 2025). We adapt this idea to structured output text classification in our compared baseline TF-GRPO-C, which is not a previously published standalone method. Our optimization ladder includes TF-GRPO-C as a rollout-based baseline, EDEO as an error-driven experience optimizer, and LEDSET as a structured experience optimizer that isolates the portion of the data that represents stable invariants from those portions representing the mutable tactics.

The evaluation of robustness of NLP has been formalized via a variety of techniques. These include behavioral reliability testing using invariant properties (Ribeiro et al., 2020), minimal perturbation contrast sets (Gardner et al., 2020), metamorphic transformation relations (Chen et al., 2018), and toolkits for analyzing different transformations, subpopulations, and adversarial attacks (Goel et al., 2021; Wang et al., 2021), alongside the broader principle of stability under perturbations rather than nominal maximization (Madry et al., 2018). Low-data intent classification provides a realistic benchmark for these conditions (Larson et al., 2019; Casanueva et al., 2020), with open-intent decision boundary modeling for out-of-scope (OOS) rejection (Zhang et al., 2021). In addition to these studies, representations of production noise such as abbreviations, misspellings, paraphrasing, and synonymy have been examined in (Sengupta et al., 2021). When few labeled examples are available,

synthetic supervision can be generated with methods such as Self-Instruct (Wang et al., 2023), Unnatural Instructions (Honovich et al., 2023), and LINGUIST (Rosenbaum et al., 2022), as surveyed in (Tan et al., 2024). However, intent-oriented augmentation is less reliable at adjacent intent boundaries (Sahu et al., 2022; Castillo-López et al., 2026).

2.2 Problem Setting

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the label space. A Language Model (LM) f_θ, provided with an inference-time context $\mathcal{E}$, predicts a label $\hat{y} = f_\theta(x, \mathcal{E})$.

Context Space Overfitting A typical unconstrained optimization objective aims to maximize nominal performance on the training set $\mathcal{D}_{train}$:

$$\mathcal{E}^* = \arg\max_{\mathcal{E}} \mathbb{E}_{(x,y)\sim\mathcal{D}_{train}} \left[\mathbf{1}(f_\theta(x, \mathcal{E}) = y)\right] \tag{1}$$

In low-data or synthetic regimes, optimizing Eq. (1) risks *context space overfitting*. Instead of distilling label semantics, the optimizer often latches onto transient lexical patterns, formatting cues, or brittle exemplar orderings (Zhao et al., 2021; Min et al., 2022). However, these solutions are highly sensitive to textual disruptions and paraphrasing. Standard test-set evaluations often fail to expose this instability, leading to an overestimation of model robustness(Gardner et al., 2020).

Robust Constrained Optimization We propose an optimization function: $\mathcal{D}_{dev}$ is a development set and $\mathcal{D}_{stress}$ is its metamorphic augmentations (e.g., noise, synonyms). Our goal is to maximize the primary task metric M (Macro-F1 in this paper) while maintaining a robustness floor τ_{stress} on the stress distribution:

$$\begin{aligned} \mathcal{E}^* = \arg\max_{\mathcal{E}} \quad & M(\mathcal{D}_{dev}; \mathcal{E}) \\ \text{s.t.} \quad & M(\mathcal{D}_{stress}; \mathcal{E}) \geq \tau_{stress} \end{aligned} \tag{2}$$

3 The Context Optimization Ladder

Equation 2 exposes structural gaps in current iterative frameworks. We view the evolution of these methods as an "optimization ladder": transitioning from success-driven sampling to error-driven rewrites, and finally to the structured discovery proposed in this work. Each step addresses a failure mode that the previous one could not resolve.

3.1 TF-GRPO-C: Success-Driven Induction

TF-GRPO-C adapts reinforcement learning principles (Shao et al., 2024; Cai et al., 2025) to favor experience with high group-relative advantage: $A_i = (r_i - \mu_G)/(\sigma_G + \epsilon)$. While effective for open-ended tasks, this method fails in classification for two main reasons:

- **No causal diagnosis:** The framework rewards lucky successes from random sampling but does not diagnose why the model fails. Without error analysis, it cannot identify or fix the "top-K confusion pairs" that drive most classification errors.
- **Signal Sparsity:** Low-data classification often results in deterministic failure. If all rollouts in a group collapse into the same wrong label, the reward variance vanishes ($\sigma_G \approx 0$). This leaves the optimizer with no signal to update the prompt, causing the process to terminate.

3.2 EDEO: Error-Driven Monolithic Optimization

EDEO bridges the signal gap by shifting from stochastic success to deterministic failure. It samples misclassified instances to guide full context rewrites. However, because it treats the experience as a *monolithic block*, it suffers from:

- **Semantic Drift:** Changes meant to fix local boundary errors often disrupt global semantic invariants. This "all-or-nothing" rewrite risks losing stable label definitions entirely.
- **Lexical Shortcuts:** Since EDEO only optimizes for development set scores ($M(\mathcal{D}_{dev}; \mathcal{E})$), it often latches onto surface-level artifacts. Without a robustness gate, these edits improve nominal performance but fail under the metamorphic noise of $\mathcal{D}_{stress}$.

3.3 Requirements for Robustness

The problems with TF-GRPO-C and EDEO show what is needed for a more robust optimizer:

1. **Separated Experience Layers:** We must keep stable task rules separate from flexible tactics. This prevents the optimization process from accidentally changing the core meaning of the labels.
2. **Small, Tested Edits:** Instead of rewriting the entire context, we use discrete `ADD` edits. Each edit must pass a check to ensure it fixes errors without damaging existing correct predictions or reducing robustness.

These requirements lead to our proposed method, LEDSET, which we describe in Section 4.

4 Layered Experience Discovery with Structured Edit Tactics

LEDSET is an iterative experience-context optimizer for the constrained objective mentioned in Section 2.2. In contrast to the previous iterative step-based optimization methods LEDSET does not require successful rollout variance. Moreover, it evaluates potential edit candidates based upon semantic preserving stress variations as opposed to the sole measure of development performance. LEDSET breaks down the inference context into 3 layers. L0 produces invariant $\mathcal{E}_{meta}$, L1 produces task and class specific semantic invariants $\mathcal{E}_{inv}$, and L2 generates mutable tactical edits $\mathcal{E}_{tac}$. With $\oplus$ denoting inference-time context concatenation, the experience context can be represented by

$$\mathcal{E} = \mathcal{E}_{meta} \oplus \mathcal{E}_{inv} \oplus \mathcal{E}_{tac}. \tag{3}$$

Only $\mathcal{E}_{tac}$ is optimized iteratively. Both $\mathcal{E}_{meta}$ and $\mathcal{E}_{inv}$ are computed once and then frozen. This paper evaluates the `ADD`-only regime, where a proposed tactic is appended to $\mathcal{E}_{tac}$ only if it passes the per-confusion-cell gate. `MODIFY` and `DELETE` are reserved for future maintenance settings and are not used in the experiments.

4.1 Stability-Plasticity Decomposition

Decomposition isolates information that will be stable from information that can change throughout the optimization process. $\mathcal{E}_{meta}$ contains specific task level constraints such as format of outputs, types of labels, while $\mathcal{E}_{inv}$ contains label meanings, ontological relations, and rules for disambiguating ambiguous terms learned from labeled examples. Statements in L1 are only stored if they continue to be true when replaced with test cases based on another example of the same label. The mutable layer, L2 with $\mathcal{E}_{tac}$, contains local tactics that are found through observing errors. While these tactics can correct boundary confusions, they cannot supersede $\mathcal{E}_{inv}$. When there exists an inconsistency between invariant L1 expressions and tactic L2 expressions, the expression in L1 prevails.

4.2 Inductive Discovery of Stable Invariants

Before tactical optimization starts, $\mathcal{E}_{meta}$ and $\mathcal{E}_{inv}$ are constructed with the following operations. L0 inspects the label space, output discipline, and task structure and extracts an instructional meta-context of the classification task at hand. As a second step, L1 uses a form of label specific deductive reasoning utilizing contrastive inference. For each label, LEDSET samples positive exemplars and negative examples from labels which are adjacent in relation to the current label. It then identifies attributes and characteristics which are common among positive exemplars but absent in negative exemplars, removes any lexical characteristics or accidents, and finally tightens the result by falsification with respect to all other labels in the dataset. The end result is a small number of disambiguating principles and semantic invariants. A summary of this process is provided in Algorithm 1.

Algorithm 1 L1 Contrastive Essence Extraction

Require: Training set $\mathcal{D}_{train}$, label space $\mathcal{Y}$

$\mathcal{E}_{inv} \leftarrow \emptyset$
for each $y \in \mathcal{Y}$ **do**
  $P_y \leftarrow \textsc{TopPrototypes}(\mathcal{D}_{train}, y)$
  $N_y \leftarrow \textsc{GetContrasts}(\mathcal{D}_{train}, \mathcal{Y} \setminus \{y\})$
  $e_y \leftarrow \textsc{ExtractEssence}(P_y, N_y)$
  $e_y \leftarrow \textsc{AbstractAndTighten}(e_y, N_y)$
  $\mathcal{E}_{inv} \leftarrow \mathcal{E}_{inv} \cup \{e_y\}$
end for
$H \leftarrow \textsc{DiscoverLabelOntology}(\mathcal{E}_{inv})$
$A \leftarrow \textsc{DisambiguationAxioms}(H, \mathcal{E}_{inv})$
return $\mathcal{E}_{inv} \oplus H \oplus A$

4.3 Error-Driven Atomic Search

After $\mathcal{E}_{meta}$ and $\mathcal{E}_{inv}$ are frozen, the tactical prompts are optimized in $\mathcal{E}_{tac}$. During each iteration, LEDSET gathers the training errors. It groups the errors based upon the confusion cells in which the errors occurred. LEDSET determines the top-K confusion pairs occurring most frequently. LEDSET performs a diagnostic examination to determine the most common failure pattern for each identified confusion cell. LEDSET provides candidate `ADD` tactics to address those failure patterns. A diagnostic examination has two parts. First, LEDSET examines each misclassified example to determine which evidence should have supported its correct label, which evidence was overweighted by the incorrect prediction, and which evidence was absent. After determining these elements for each misclassified example, LEDSET compiles this information for each confusion cell to formulate a hypothesis regarding which type of local tactic should be used.

Algorithm 2 illustrates the ADD-only optimization loop used in this work. E_{err} denotes the misclassified training examples under the current experience $\mathcal{E}$, and E_{cell} denotes sampled errors from a selected confusion pair. The diagnosis d summarizes missed gold-label evidence, overweighted wrong-label evidence, and absent cues. $\mathcal{C}$ is the set of proposed ADD edits. The selected edit c^* maximizes $\mathrm{F}(c, E_{\text{cell}}) - \lambda\,\mathrm{B}(c, E_{\text{cell}})$, where F counts fixed errors and B counts broken cases, and equal scores are tie-broken with stress variants generated from E_{cell}.

4.4 Metamorphic Robustness via Structural Design and a Per-Edit Gate

LEDSET achieves metamorphic robustness through both structural and gating mechanisms. At the structural level, L1 discovers invariants using abstraction, falsification, and replacement tests. However, L2 edit operations can never alter $\mathcal{E}_{meta}$ or $\mathcal{E}_{inv}$. Gating occurs locally, with each candidate tactic being evaluated based only upon the confusion cell it is meant to repair. The confusion-cell gate determines the best candidate tactic by evaluating all candidates as

$$c^* = \arg\max_{c \in \mathcal{C}} \Big(\mathrm{F}(c, E_{cell}) \; - \; \lambda \cdot \mathrm{B}(c, E_{cell}) \Big), \tag{4}$$

$$\text{accept } c^* \iff \mathrm{F}(c^*, E_{cell}) > 0, \tag{5}$$

where E_{cell}, $\mathcal{C}$, $\mathrm{F}(c, E_{cell})$, $\mathrm{B}(c, E_{cell})$, and λ denote examples from the targeted confusion cell, proposed ADD edits, examples fixed by adding c, examples broken by adding c, and the breakage penalty, respectively. A larger value of λ increases the penalty for tactics that fix ambiguous cases at the cost of breaking previously correct classifications.

A set of stress test data $\mathcal{D}_{stress}$ is generated through semantics-preserving transformations on held-out development data, following behavioral testing, metamorphic testing, Robustness Gym, and TextFlint (Ribeiro et al., 2020; Chen et al., 2018; Goel et al., 2021; Wang et al., 2021). Transformation types, detailed in Appendix B, include keyboard noise, mixed-intent fillers, adversarial prompt injection, speech-to-text errors, irrelevant context, token deletion, and disfluency injection. These stress tests evaluate whether the optimized context remains stable under surface changes while preserving the original input intent. This evaluation differs from robust optimization over model parameters (Madry et al., 2018), since it examines updates to contextual information.

Algorithm 2 LEDSET ADD-only Regime Used in This Work

Require: $\mathcal{D}_{train}$, labels $\mathcal{Y}$, iterations S, top confusions K, proposals N, break penalty λ

$\mathcal{E}_{meta} \leftarrow$ DISCOVERL0$(\mathcal{D}_{train})$; $\mathcal{E}_{inv} \leftarrow$ EXTRACTL1$(\mathcal{D}_{train}, \mathcal{Y})$
$\mathcal{E}_{tac} \leftarrow \emptyset$; $\mathcal{E} \leftarrow \mathcal{E}_{meta} \oplus \mathcal{E}_{inv} \oplus \mathcal{E}_{tac}$
for $s = 1$ **to** S **do**
  $E_{\text{err}} \leftarrow$ GETERRORS$(\mathcal{D}_{train}, f_\theta, \mathcal{E})$
  if $E_{\text{err}} = \emptyset$ **then**
    break
  end if
  for $p \in$ TOPKCONFUSIONS(E_{err}, K) **do**
    $E_{\text{cell}} \leftarrow$ SAMPLE(E_{err}, p)
    $d \leftarrow$ DIAGNOSE(E_{cell})
    $\mathcal{C} \leftarrow$ PROPOSEADD(d, E_{cell}, N)
    $c^* \leftarrow$ SELECTBESTADD$(\mathcal{C}, E_{\text{cell}}, \lambda)$
    if $\mathrm{F}(c^*, E_{\text{cell}}) > 0$ **then**
      $\mathcal{E}_{tac} \leftarrow \mathcal{E}_{tac} \cup \{c^*\}$; $\mathcal{E} \leftarrow \mathcal{E}_{meta} \oplus \mathcal{E}_{inv} \oplus \mathcal{E}_{tac}$
    end if
  end for
end for
return $\mathcal{E}$

5 Experimental Setup and Results

Two low data intent text classification settings are evaluated. The first one is a public benchmark called CLINC150 (Larson et al., 2019) and the second one is a created ultrasonic NonDestructive Testing (NDT) digital twin intent routing dataset. The experiments are designed around the following research questions. RQ1: Do iterative experience context generation and modification techniques have better overall performance over fixed artifact prompt optimization methods? RQ2: Does error-driven techniques outperform rollout success-based optimization for deterministic classification decisions? RQ3: Do the design choices in LEDSET such as auto-discovered $\mathcal{E}_{meta}$, $\mathcal{E}_{inv}$ plus confusion cell-focused edits in L2 decrease the context-space overfitting? RQ4: Does LEDSET contribute additional value on top of the baseline LMs after LoRA finetuning?

5.1 Experimental Setup

The five CLINC150 subsets to be chosen for benchmark evaluation are banking, work, travel, utilities,

and small talk. The ultrasonic NDT dataset has 1000 examples, of which 600 synthetically and 400 organically labeled by an ultrasonic testing expert. The user queries are generated synthetically and the labels are assigned by the expert. Outputs from all datasets have a structured output format that includes three fields one field to store the intent label, one field to store a confidence score between 0 and 1, and a third field to include a brief text description of why the model made its classification decision. An output that does not obey the structure is counted as misclassification. The ultrasonic NDT dataset contains eight digital twin-related intent classes, one general conversation class, and one OOS label, detailed in Appendix C. These labels are inspired by industrial Asset Administration Shell (AAS) semantics, a standard maintained by the Industrial Digital Twin Association (IDTA; IEC 63278) (Industrial Digital Twin Association, 2025; International Electrotechnical Commission, 2023), with digital twin usage in NDT illustrated in (Yallıç et al., 2022; Leinenbach et al., 2025). The methods to be compared fall under two categories of prompt optimization. The first category, which is called the "fixed artifact" category consists of MIPROv2, GEPA, COPRO, and APO/ ProTeGi. The second category called the iterative experience context optimization consists of TF-GRPO-C, EDEO and LEDSET. The baseline model is the Qwen3 LM family of models with 4 billion parameters and 8-bit quantization. The specific model name that we chose is `Qwen3-4B-Instruct-2507-FP8` (Yang et al., 2025; Qwen Team, 2025). Rather than being deployed as a competing method of optimizing prompts, SFT with LoRA is used as a base model for deploying optimized prompts. It is implemented using the unsloth library (Han et al., 2023). All optimized prompts are frozen during inference. Both comparisons utilize the same 4-fold stratified cross validation protocols. Train, development and test splits are 60, 15, 25, respectively. A single seed value of 4242 42 is used for fold splits, and we report cross-fold standard deviation as our variance estimate. Multi-seed analysis is left to future work. Optimization is done only on the training sets of each fold. Performance results are reported after each optimization iteration using the reserved development sets of each fold. Since classifications are generated by the finetuned version of the model and proposals are generated by the non-finetuned version of the model, the improved performance of the SFT +LEDSET configuration over SFT comes from better exploration of the prompt space rather than modifying model parameters. `Qwen3-4B-Instruct-2507-FP8` model is the only proposer LM in the study for fair comparison. The primary metrics are MACRO F1 since it captures class imbalances. Evaluation of robustness is conducted through Stress MACRO F1 on $\mathcal{D}_{stress}$. Optimization and inference costs are measured in tokens. The total optimization token budget for each method is computed by summing the input and output tokens spent for proposer, evaluator, and candidate rewrite API calls across all cross-validation folds and multiple datasets in the case with CLINC150 subsets. The tokens in evaluation stage using the test set are excluded.

5.2 Results

Figure 1 shows test performance on the left side and stress performance on the right side. Each selected CLINC150 domain is evaluated as an independent intent classification task, in which the number of classes and examples extensively explained in Appendix A. Horizontal dashed lines represent mean model MACRO F1 without prompt optimization because these models do not spend optimization tokens.

The seven prompt optimization methods analyzed in this study are MIPROv2, GEPA, COPRO, APO / ProTeGi, TF-GRPO-C, EDEO, and LEDSET with L0, L1, and L2 stages. For APO, TF-GRPO-C, EDEO, and LEDSET, the iteration parameter is set to 2, 4, or 8. COPRO uses depth values 2 and 4. MIPROv2 and GEPA use light and heavy automatic modes. The remaining configurations are mentioned in Appendix B. All prompt optimization techniques are run on top of the `Qwen3-4B-Instruct-2507-FP8` model, while LEDSET is additionally run on LoRA SFT version of `Qwen3-4B-Instruct-2507-FP8` model. Besides, GPT5.1 model without reasoning is evaluated as proprietary reference baseline (OpenAI, 2025a,b).

In the test performance plot, DSPy methods, namely MIPROv2, COPRO, and GEPA, appear as low-cost prompt optimizers because all of them use less than 10^8 summed optimization tokens. However, their MACRO F1 scores remain lower than the iterative experience-based optimization methods. Among the DSPy methods, GEPA performs best and is comparable to APO. However, APO with iteration value 4 has a slight advantage over

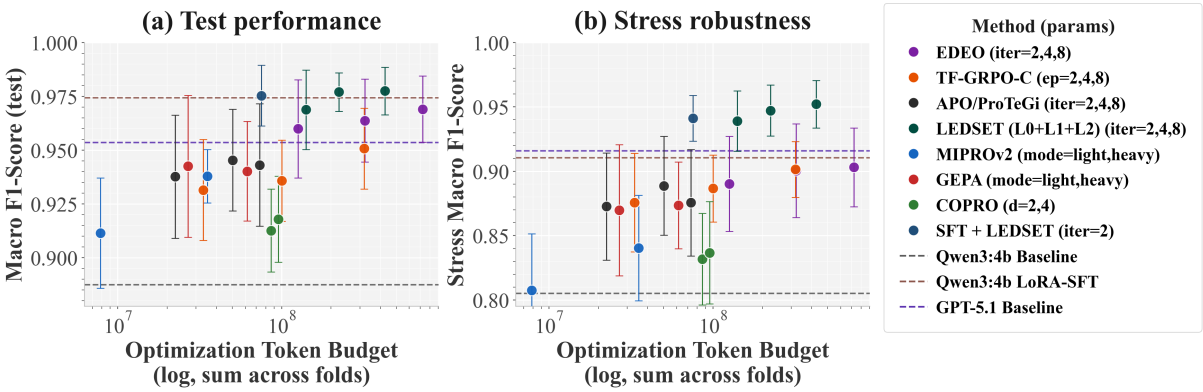

Figure 1: CLINC150 subsets of utility, work, small talk, banking, and travel (a) Test set MACRO F1 score and (b) Stress MACRO F1 performance plotted against the total number of used optimization tokens for each prompt optimization method

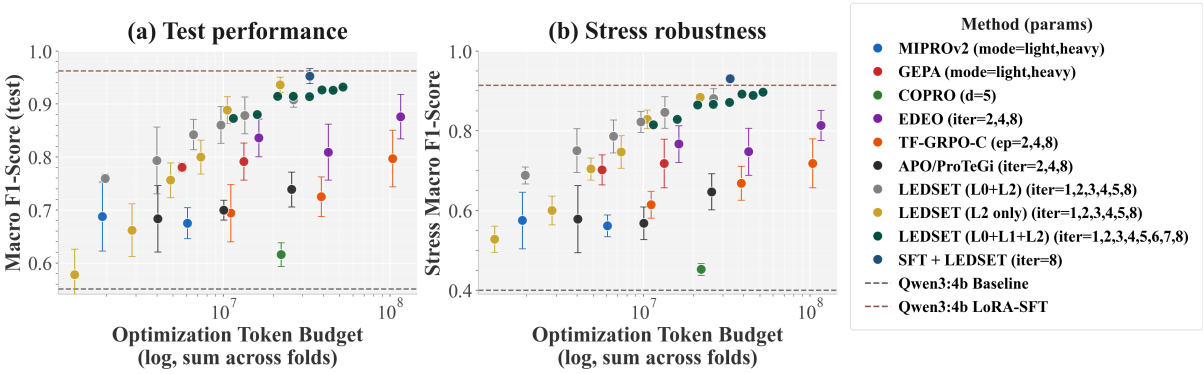

Figure 2: Ultrasonic NDT digital twin intent classification dataset (a) Test MACRO F1 score and (b) Stress MACRO F1 performance plotted against the total number of used optimization tokens for each prompt optimization method

GEPA. Under the token budget of 10^8, APO is the strongest method.

Above the token budget of 10^8, LEDSET with L0+L1+L2 achieves the highest performance. The second-best method is EDEO, followed by TF-GRPO-C. LEDSET with iteration values 4 and 8 exceeds 97.5% nominal mean MACRO F1 and has the smallest standard deviation. Thus, LEDSET is the most stable method across the selected datasets. EDEO also obtains above 96% mean test MACRO F1 for all three configurations. At iteration 8, TF-GRPO-C reaches slightly above 95% mean MACRO F1. Only configurations that surpass the GPT5.1 and SFT models are LEDSET with 4 and 8 iterations, and SFT+LEDSET with 2 iterations. Furthermore, only method other than LEDSET that surpasses GPT5.1 is EDEO regardless of the iteration numbers. Additionally, SFT+LEDSET achieves comparable scores with LEDSET spending substantially fewer optimization tokens due to having a much lower training set error rate.

Further investigating the stress robustness subplot (b), the difference between the performances of LEDSET and all the other prompt optimization techniques are widened. While the increase in the LEDSET iteration numbers do not bring additional nominal performance improvement in (a), compute is positively traded for robustness in (b). TF-GRPO-C with 8 epochs and EDEO with 4 iterations have comparable scores above mean 90% stress MACRO F1 with similar token budgets costs. In stress evaluations, LEDSET and SFT+LEDSET exceed GPT5.1 and SFT baselines. APO/ProTeGi rises as the most efficient and effective fixed-prompt optimization technique but all of them are under GPT5.1 and LoRA-SFT baselines as in (a). However, different than (a), EDEO falls behind of GPT5.1 and LoRA-SFT in stress tests. Whereas LoRA-SFT model overperforms GPT5.1 in nominal test performance (a), it falls slightly short in metamorphic transformation results. Moving from TF-GRPO-C to EDEO and then to LEDSET improves nominal MACRO F1 at increasing token cost, so the iterative experi-

ence optimizers exceed the fixed-artifact ones but at higher optimization cost, supporting the ladder of Section 3.

Figure 2 visualizes a similar nominal test and stress test performance investigation using seven context optimizers but with Ultrasonic NDT intent classification dataset based on human-digital twin interactions. In addition, an ablation study for LEDSET's different layers are carried out. Token costs are increasing from LEDSET with only L2 to LEDSET with L0+L2, and to LEDSET with L0+L1+L2. In test performance (a), LEDSET with L2 only surpasses its L0+L1+L2 counterpart even with much lower token budget. However, in (b), it is seen that L2 overfocused on local confusion pair relationships without understanding more generalizable stable semantics. Thus, stability-plasticity decomposition using L1 and L2 provides meaningful performance gains.

Comparing all the best scores between the fixed-prompt and experience optimizer techniques without considering the token budget, all iterative experience learning methods go beyond fixed-artifact counterparts. It is observed that utilizing different configurations of LEDSET without SFT, more token budget yields better stress robustness (b). Moreover, a configuration of LEDSET is always more efficient than all other methods for all comparable token budgets (a, b). LoRA-SFT outperforms all the optimization techniques in test performance (a) but SFT+LEDSET with 8 iterations rises as a winner against all other experimented techniques in terms of metamorphic robustness evaluation (b). The test-to-stress MACRO F1 gap averages 2.8% for LEDSET, 6.3% for APO and 10.1% for MIPROv2 on CLINC150. On the ultrasonic NDT to 4.5% for LEDSET, 7.6% for GEPA, 10.9% for APO, and 15.4% for COPRO. Per-transformation accuracy values are reported in Appendix C. They show that LEDSET and SFT+LEDSET are strongest especially under injection and mixed-intent transformations, where semantic distraction is highest.

6 Conclusion

The results in this study indicate that iterative experience optimization methods outperform fixed artifact prompt optimization methods in both nominal MACRO F1 and Stress MACRO F1. For analyzing the improvements in experience context generation, an optimization ladder is proposed, which involves TF-GRPO-C, EDEO and LEDSET. The main limitation in TF-GRPO-C is that it relies on successful sampled rollouts. Although EDEO improves this direction by diagnosing errors directly, it still optimizes a monolithic experience representation. Current last step in the ladder, LEDSET separates stable semantic invariants from mutable tactical edits.

LEDSET achieves the strongest overall robustness, considering the prompt optimization methods without SFT. The decomposition strategy of L0, L1 and L2 isolates highly abstract level classification task constraints, stable label semantics, and local confusion-cell tactics, respectively. Thus, label-specific boundary errors are corrected without enabling local tactics to override more generalizable semantic rules. The ablation results show that L2-only optimization can improve nominal performance, but it can overfocus on local confusion pair relationships. However, separated L1 and L2 improves stress test robustness and therefore, decreases context-space overfitting. The lowest test-to-stress MACRO F1 performance gap occurred with LEDSET on all datasets.

The metamorphic transformation results support the same deduction in more detail. For CLINC150, LEDSET with L0+L1+L2 obtains the highest or tied highest accuracy value in all seven metamorphic transformation families. For the ultrasonic NDT dataset, SFT+LEDSET obtains the highest or tied highest value in six of the seven transformation families. The largest gains appear for injection and mixed intent transformations, since these transformations require preserving the intended classification decision under adversarial or semantically distracting additions. Moreover, LEDSET contributes additional value on top of LoRA finetuning, showing that context optimization can complement parameter-efficient finetuning.

The scope of these findings is limited to low data, single label, monolingual English intent classification with a fixed open-weight model family and the `ADD`-only version of LEDSET. Stress MACRO F1 should be interpreted as robustness under the evaluated metamorphic transformation families, not as robustness under all possible distribution shifts. This work does not include multi-seed reruns, pairwise significance testing, confidence intervals, or human evaluation of the discovered L1 invariants and L2 tactics. Future work will extend LEDSET with `MODIFY` and `DELETE` operators for evolving datasets.

7 Limitations

The empirical findings of this research apply to an exceedingly narrow operational range of low amount of datasets with single labels as output, using a fixed open-weight model family adapted either with or without LoRA and tested under the ADD-only version of LEDSET.

The largest single constraint is that we have examined only the additive, iterative regime , where new tactics are added to $\mathcal{E}_{tac}$ at each cell's first acceptance gate of Eq. 5). However, if there is an issue with the added tactic in the future iterations, the previously accepted tactics can never be modified, deleted or replaced. The runtime gates for adding stress $\mathcal{D}_{stress}$ and development $\mathcal{D}_{dev}$, as described in the larger context of the LEDSET framework, are also not considered in the ADD-only version. Isolating these two mechanisms allow us to measure the structural impact of layered discovery combined with localized atomic repairs, independent of the full rewrite processes involved. Yet, our results describe the construction of context in the bootstrapping phase and do not address how to maintain context over longer periods.

The MODIFY and DELETE operators described in Section 4 are specifically designed to handle accumulation, interaction and staleness of $\mathcal{E}_{tac}$ in evolving datasets with time. The effect of such complexity are not assessed in this work. In addition to the single industrial domain, we report on five CLINC150 subsets and one industrial Ultrasonic NDT routing example which uses Asset Administration Shell semantics. Specifically, we use a single industrial domain, a single ontology family and a specific type of operator language. Furthermore, we only assess the select CLINC150 subsets instead of assessing all CLINC subsets due to compute requirement for this work. The factorization $\mathcal{E} = \mathcal{E}_{meta} \oplus \mathcal{E}_{inv} \oplus \mathcal{E}_{tac}$ is particularly well-suited to deterministic, bounded, and single label classification problems. More specifically, we have also not tested multi-label, hierarchical or large label space problems, streaming label discovery, multi-turn dialogue state tracking, and long document cases. As $|\mathcal{Y}|$ increases, the competing use of the context budget by L1 and $\mathcal{E}_{tac}$, as well as the need to replace the lossless concatenation in Eq. 3 with a retrieval-conditioned version will eventually need a retrieval-conditioned variant.

The robustness evidence relies upon a metamorphic stress test set that are generated using all training cases for which there are eight hand-coded families of transformations similar to CheckList, metamorphic testing, Robustness Gym, and TextFlint. Stress MACRO F1 does not measure how well the model performs when placed on an entirely different distribution. Therefore, it will never cover all possible concept drifts, such as multilingual paraphrases and possible institutional specific failures, while the catalog can provide some coverage of these types of failures in the existing transformation families. Therefore, strong evidence of Stress MACRO F1 should be interpreted as evidence that the model has not overfit to a particular subset of the space of contexts that are tested with respect to the perturbations used during training. However, this does not constitute evidence that the model will exhibit all potential deployment-invariant behaviors.

The algorithm presented in Algorithm 1 also assumes that for each label, there exists a unique semantic content that can be induced from a positive prototype relative to a negative prototype. This assumption is violated whenever there exist labels that serve as OUT_OF_SCOPE or GENERAL_CONVERSATION), whose definition depends on what they exclude, and not on their commonality of signature among positives. Furthermore, this violation is exacerbated by the presence of generator artifacts in large amounts of synthetic data. While abstraction and falsification may help to mitigate the effects of this type of artifact, they will not completely eliminate them. In addition, because the L1 override takes precedence over the L2 override, a failed invariant will propagate to every downstream tactical edit. Hence, this represents a true cost associated with prioritizing semantic stability over local repair.

The results of Section 5 can be considered as observations from a single hybrid benchmark, one LoRA recipe and one non-finetuned proposer configuration. The protocol predicts using LoRA adapted base model to propose using unadapted base models. This protocol will restrict the claim that the benefits of optimized contexts are realized while enabling further investigation into possible alternative combinations such as optimizing other components in the context, i.e., how does the combination change with better base models, larger LoRA capacity, or additional organic training signals and/or for specific application domains, where the post-SFT residuals do not appear to be localized at interpretative semantic boundaries.

LEDSET is the least expensive optimizer among those included in our experience family in cost measurement but not inexpensive in absolute terms, since discovery and proposal generation both depend on multiple calls to an LLM. For the purposes of this study a single global seed, including 42, is used for fold creation, optimizer internal randomness and SFT initializations. Confidence intervals are not calculated nor did we conduct pair-wise significance testing between the top performing optimizers (i.e., EDEO, APO, and LEDSET). Thus, readers should consider the lack of multi-seed rerun data when assessing the magnitude of the reported gaps as they relate to the order of optimization performance.

All of the benchmarking methods are monolingual English. The L0 schema, the L1 Contrastive Essence Extraction (Algorithm 1), stress transformation catalog, and all proposer prompts are written in monolingual English. Therefore, we cannot assume that the inductive L1 and transformation robust behavior will automatically be applicable with no modifications to typologically distant languages, input data with code mixtures, low resource environments or a prior environment where the proposer's prior is weaker. CLINC150 has been public since 2019 (Larson et al., 2019). Therefore, it is highly likely the Qwen3 router and proposer have seen this task many times over before they were trained. Because contamination is uniform across all methods that use the same inference base, we believe there are no issues with relative nominal MACRO F1 based on within paper comparisons. The nominal to stress MACRO F1 channel is much more robust than nominal because the transformations produce surface forms which the pretraining corpus never saw together as a single label. Thus, the nominal to stress generalization gap cannot be inflated by contamination. Finally, the LMs do not have access to the ultrasonic NDT benchmark and, therefore, this issue does not apply to it.

A fundamental principle behind the LEDSET method is that the equation $\mathcal{E} = \mathcal{E}_{meta} \oplus \mathcal{E}_{inv} \oplus \mathcal{E}_{tac}$ results in auditable output. Found L1 essences of what labels have in common and L2 tactic of locally understandable repairs are main concepts. To determine if this is true, we test it indirectly via our MACRO F1 and stress MACRO F1. However, we did not perform a human evaluation to determine if the L1 statements are deemed accurate, if the L2 tactics that are accepted are reasonable, and/or if they would have been chosen instead of the single, unstructured experience provided by EDEO. The structural statement should be viewed as being backed by robustness characteristics vs. an actual auditing of accuracy.

The edits are strictly local in LEDSET due to the operation of $\mathrm{F}(c, E_{\text{cell}}) - \lambda\,\mathrm{B}(c, E_{\text{cell}})$. Therefore, the edits will be limited in scope as they cannot modify or replace strategies globally as in EDEO(), which can mask strategy improvement for global development with strategy decline along boundaries but the edits could also be rejected if they are only beneficial through future interactions. Moreover, the choice of λ depends on datasets. Too small values permit brittle trades and too large λ are potentially useful in error-dense regions. In this paper, we report a single λ operation and, hence, we do not characterize the full sensitivity surface.

8 Broader Impact and Ethics

There exist some applications where decision support tools based on low-data and/or low-budget text classification have been developed to assist in making decisions with significant stakes. Any time there is an error made within this type of classification, it has the potential to create an adverse impact in the downstream operational decisions. Here, our objective is to enhance the performance of the classifier by minimizing such errors. Secondly, we try to minimize the risk that errors will occur due to minor surface form variations or simply due to intentionally crafted adversarial phrasing. Important disclaimer here is that this paper does not replace human oversight in safety-related applications. Data generated synthetically to train or test classifiers can include stylistic artifacts or hidden biases that are created through the use of the model creating synthetic data. Our work demonstrates does not demonstrate all possible risks. A secondary advantage of using structured experience optimization is the enhanced ability to audit. Updates made to a system at the tactical level are easier to understand and review than single prompt updates. Structured ways of representing knowledge are much more amenable to auditing or red-teaming than unstructured representation methods. However, before utilizing any method that increases either the intent-routing or classification abilities in a manner, in which they will be applied to real world applications, keep in mind that LMs will generate decisions and those decisions could potentially have a positive or negative effect

on users. Thus, we propose a number of practical safeguards for developers, such as constraining contextualization to inference time. reporting failures clearly, and being conservative when evaluating the robustness of an approach.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2026. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*. Oral presentation.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Galo Castillo-López, Alexis Lombard, Nasredine Semmar, and Gaël de Chalendar. 2026. How DDAIR you? disambiguated data augmentation for intent recognition. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 274–286, Rabat, Morocco. Association for Computational Linguistics.

Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. 2018. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys*, 51(1).

DSPy. 2026a. DSPy documentation: COPRO. `https://dspy.ai/api/optimizers/COPRO/`. Documentation page. Accessed: May 13, 2026.

DSPy. 2026b. DSPy documentation: MIPROv2. `https://dspy.ai/api/optimizers/MIPROv2/`. Documentation page. Accessed: May 13, 2026.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 7 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Daniel Han, Michael Han, and Unsloth team. 2023. Unsloth.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*. Poster.

Industrial Digital Twin Association. 2025. Specification of the asset administration shell part 1: Metamodel. Technical Report IDTA-01001-3-1-2, Industrial Digital Twin Association. Version 3.1.2.

International Electrotechnical Commission. 2023. IEC 63278-1:2023 asset administration shell for industrial applications — part 1: Asset administration shell structure. International Standard.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into self-improving pipelines. In *International Conference on Learning Representations*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Frank Leinenbach, Dharma Panchal, Cemil Emre Ardic, Marina Klees, Michael Peters, and Florian Römer. 2025. Digital representation and networking of nde information and systems. *Research and Review Journal of Nondestructive Testing*, 3(2).

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon,

Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2025a. Gpt-5.1: A smarter, more conversational chatgpt. https://openai.com/index/gpt-5-1/. Accessed: 2026-05-25.

OpenAI. 2025b. Gpt-5.1 instant and gpt-5.1 thinking system card addendum. https://openai.com/index/gpt-5-system-card-addendum-gpt-5-1/. Accessed: 2026-05-25.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

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, Singapore. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3-4B-Instruct-2507-FP8. https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507-FP8. Model card. Accessed: May 20, 2026.

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, Haozhu Wang, Han Ding, Yuzhe Lu, Zhichao Xu, Yun Zhou, Balasubramaniam Srinivasan, Qiaojing Yan, Yueyan Chen, Haibo Ding, and 2 others. 2025. A systematic survey of automatic prompt optimization techniques. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33078–33110, Suzhou, China. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022. LINGUIST: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 218–241, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Sailik Sengupta, Jason Krone, and Saab Mansour. 2021. On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 68–79, Online. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui

Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, and 15 others. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024. PromptAgent: Strategic planning with language models enables expert-level prompt optimization. In *International Conference on Learning Representations*. Poster.

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, Toronto, Canada. Association for Computational Linguistics.

Fatih Yallıç, Özlem Albayrak, and Perin Ünal. 2022. Asset administration shell generation and usage for digital twins: A case study for non-destructive testing. In *Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2022)*, pages 299–306. INSTICC, SciTePress.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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*. Poster.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

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 *International Conference on Learning Representations*. Poster.

A Dataset Details

The data used for all experiments includes each of the single-label English intent classification dataset, with each example represented as a two-record format (the user's message and the assistant's JSON label). Required fields within the output include the class label and a confidence value in $[0, 1]$ along with a short reason why the given response was selected. Any outputs which cannot be interpreted to fit the anticipated JSON structure will count towards the number of errors made.

Each of the five domains within the CLINC150 (Larson et al., 2019) is treated separately as its own intent classification task. The domains were developed by taking the `clinc_oos` plus training split and splitting it among the five domains. In each of these domains there will be 15 in-domain intents plus `oos`. There are 100 examples for each in-domain intent and 250 examples for `oos`, this gives us 1750 total examples and 16 possible labels across the five domains.Inputs are wrapped as `The user's query: <utterance>`. The assistant output key is `intent`.

The applied dataset contains 1000 ultrasonic nondestructive testing (NDT) user queries for routing an assistant connected to an Asset Administration Shell (AAS)-style digital twin. The output key is `conversation_type`. The label set contains eight digital-twin and NDT work labels plus `GENERAL_CONVERSATION` and `OUT_OF_SCOPE`. The released validation report records zero structural failures. It also records 37.0% context-bearing rows, 31.7% advanced-modality rows, 20.5% standards-referenced rows, 55.1% high confidence rows, 20.4% low-band confidence rows, and 10.4% very low confidence rows.

Evaluation splits. For every dataset, the experiment runner applies 4-fold stratified cross-validation with seed 42. Within each fold, optimization uses only the training portion. The inner development split is used for optimizer selection and stress-set construction, while the fold test split is used for final nominal and stress evaluation.

B Hyperparameters and Experimental Configuration

We write the details about the experiment configuration details described in Section 5, below. The public benchmark consists of the five CLINC150 subsets: banking, work, travel, utility and small talk. The applied benchmark is the Ultrasonic NDT Benchmark with 1000 examples created with a hybrid manner. It consists 600 synthetic and 400 expert-authored data rows. All experiments reported within this paper were performed on 4-fold stratified cross-validation with shuffle enabled. A single fixed global random seed number of 42 is applied, which is responsible for fold construction, internal stochasticity of the optimizers, and initializations of SFT. Additionally, because there is no averaging across multiple random seeds we provided a comment regarding variance in Section 7. All optimizer iteration steps are constrained to be generated from training data available only to each specific fold. The test fold remains unaltered. The development split in folds is used for both selecting candidates during the inner loop and constructing stress test sets. The base system prompt that is specifically modified for the datasets is loaded automatically. The base system prompt includes the classification task, all the labels, and the structured output design for the purpose of aligning the language model with the task while providing the minimum knowledge. An example for CLINC150 banking subset is here:

> Classify the user's banking intent and output only JSON with three keys below. **intent**: `freeze_account`, `routing`, `pin_change`, `bill_due`, `pay_bill`, `account_blocked`, `interest-rate`, `min_payment`, `bill_balance`, `transfer`, `order_checks`, `balance`, `spending_history`, `transactions`, `report_fraud`, `oos`; **confidence**: float value between 0 and 1; **reasoning**: $1-2$ sentences.

We employ currently only one Qwen3-4B-Instruct family model of `Qwen3-4B-Instruct-2507-FP8`. Its decoding is deterministic with a temperature of 0.0, a maximum of 2048 tokens permitted for the JSON style output to fit inside, and a 120s timeout. Because decoder outputs are deterministic when executing inference via router, we do not turn on either `top-p` or `top-k`. We use the default stop tokens defined by the framework for the structured output regime. We do not define any custom string values to signal termination of generating output. A second independent critic or scorer model is not used. There are two scoring points in time. The first one is at the gate in (Eq. 4) and the second one at the EDEO development selection point using

Subset	Examples	Labels	Intent labels
Banking	1,750	16	freeze_account, routing, pin_change, bill_due, pay_bill, account_blocked, interest_rate, min_payment, bill_balance, transfer, order_checks, balance, spending_history, transactions, report_fraud, oos
Work	1,750	16	pto_request_status, next_holiday, insurance_change, insurance, meeting_schedule, payday, taxes, income, rollover_401k, pto_balance, pto_request, w2, schedule_meeting, direct_deposit, pto_used, oos
Travel	1,750	16	plug_type, travel_notification, translate, flight_status, international_visa, timezone, exchange_rate, travel_suggestion, travel_alert, vaccines, lost_luggage, book_flight, book_hotel, carry_on, car_rental, oos
Utility	1,750	16	weather, alarm, date, find_phone, share_location, timer, make_call, calculator, definition, measurement_conversion, flip_coin, spelling, time, roll_dice, text, oos
Small talk	1,750	16	who_made_you, meaning_of_life, who_do_you_work_for, do_you_have_pets, what_are_your_hobbies, fun_fact, what_is_your_name, where_are_you_from, goodbye, thank_you, greeting, tell_joke, are_you_a_bot, how_old_are_you, what_can_i_ask_you, oos

Table 1: CLINC150 domain subsets used in the experiments. Each subset is evaluated independently.

Label	Count	Meaning
AAS_OPERATION	101	Execute a state change or device/session action.
AAS_QUERY	95	Retrieve a live value without judgment or persistent watch.
RAG_QUERY	92	Ask for documentation, theory, standards, or reference knowledge.
AAS_METADATA	94	Ask about twin structure, freshness, mapping, schema, or capability.
AAS_VALIDATION	141	Check whether a state satisfies criteria or limits.
AAS_MONITORING	102	Start, modify, resume, cancel, or inspect persistent tracking.
AAS_DIAGNOSTICS	98	Triage faults, alerts, symptoms, or likely causes.
AAS_RECOMMENDATION	104	Ask what settings, actions, or next steps are advisable.
GENERAL_CONVERSATION	86	Social or filler content without technical routing intent.
OUT_OF_SCOPE	87	Non-domain, unsafe, fraudulent, or unsupported requests.

Table 2: Ultrasonic NDT intent labels and counts in the released 1,000-row dataset.

the active router LM. We have configured reliability settings to include a retry budget of 3, an exponential backoff interval of 1s to 30s, and an error policy of continuing, where the fatal errors during the API calls do not finish the whole process. The proposer is frozen and defaults to the active router LM unless overridden, and utilizes temperature $= 0.7$, top-p=1.0, maximum tokens $= 8000$, and a timeout of 300s. The proposer has never been finetuned even if it is stacked atop an SFT-based architecture in Section 5.5. We have configured stress set evaluations. We first evaluate against the full development set. Then, we evaluate against seven transformation families that are currently active and prune to extension in the future for more robustness areas. Finally, we have enabled checkpointing and token usage. LLM caching by default. Moreover, the parallel API calls are configured so that we maximize the usage of the available GPUs for all parameter sweep scripts in the experiments.

SFT and LoRA backdrop. The SFT overlay uses the Unsloth (Han et al., 2023) and TRL frameworks. We use Qwen3-4B-Instruct-2507-unsloth-bnb-4bit as the base model, with the train_on_responses_only configuration. For all the datasets, we run a 5-fold stratified cross-validation with a fixed random seed of 42. The learning rate is set to 5×10^{-5} using an adamw_8bit optimizer and a cosine scheduler (10 warmup steps). We use an effective batch size of 8, weight decay of 0.01, and bf16 or fp16 precision. For model selection, we use early stopping with a patience of 5 evaluations.

The LoRA adapter use rank $r = 16$, $\alpha = 32$, dropout 0.0, and targeted q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj. The CLINC150 subsets uses a 512 max sequence length and 1 epoch. The Ultrasonic NDT data uses a 1024 max length and 2 epochs.

The baseline LM uses the dataset-specific base

system prompt without additional optimization. The model is used directly for the input output sequence after the model weights are available. MIPROv2 and GEPA methods have their `auto` mode parameter that can take values of `light`, `medium`, and `heavy`. We have used only `light` and `heavy` modes for both techniques in the paper. Development fraction is 15%. MIPROv2 has 3 bootstrapped demos and 3 labeled demos. The temperature is set to 0.7. GEPA has 8 threads and fallback budgets of 300 metric calls and 20 full evaluations. Its reflection LM temperature is 0.7. COPRO uses breadth 8 and has two different depth values of 3 and 5. All three DSPy approaches use prompt maximum tokens of 2048 for the classification output. APO (ProTeGi) has three different iteration numbers, namely 2, 4, 8. It has a beam width of 4, and 4 textual gradients for each iteration. Furthermore, it has 1 candidate per gradient, 2 Monte-Carlo paraphrases for stepped prompts, max expansion factor 8 and the minibatch size is defined as 64. Inner development split is again 20%. The critique and rewrite LM temperatures are fixed at 0.7 with maximum token budgets of 1024 and 8000, respectively. In addition, exploration constant $c = 1.4$, and early-stop patience is 2.

TF-GRPO-C configuration parameters are as follows: Epoch numbers of 2, 4, and 8. Batch size is 50. Moreover, rollout group size 25, rollout temperature 0.7, max rules 20, and development split of 20% are additional details about the configuration. Furthermore, the numerical stability epsilon is fixed to 10^{-8}. EDEO also uses 2, 4, and 8 iterations. The candidate group size is set to 10. Additional parameters for EDEO are, top-K memory 6, development fraction 15%, error sample size 14, observed error examples by the proposer LM 10, memory sample size 3, and an early stop mechanism after 4 iterations, if there are no improvements from the previous iteration. Rewrite candidates inherit the global proposer decoding setting as temperature is equal to 0.7 and max tokens set to 8000. The scoring rule for every proposed candidate is designed on the fold-local development split and accepts the argmax over accuracy and MACRO F1. Potential ties are broken in favor of MACRO F1. Only if there is a strict improvement on the tuple, the proposal is accepted. LEDSET is configured in its `ADD`-only regime. The iteration numbers are $S = 2, 4, 8$, top confusion pairs per iteration $K = 2$, per-confusion sample size 5, diagnosis error sample size 10, `ADD` proposals per pair $N_{\text{prop}} = 6$, and the gate break penalty $\lambda = 1.0$. L1 prototypes per label is set to 20 and negative examples per adjacent label are 5. The discovery temperature is 0.7, and discovery max tokens 8000. Candidate ranking uses the score of `F` $-\lambda\cdot$ `B`, and acceptance requires `F`> 0. If the scores are tied for multiple proposal candidates, then, the tied ones are testes on a stress test with 20 generated stress examples for the confusion pair.

C Metamorphic Transformation Families

The Stress Set $\mathcal{D}_{stress}$ measures whether the classification model maintains its original class on input transformations that do not affect the class (i.e., label preserving). For every source example (x, y) there exists a transformation m, producing $(m(x), y)$. The true label is never modified. The catalog utilizes Behavioral and Metamorphic Testing methods (Ribeiro et al., 2020; Chen et al., 2018; Goel et al., 2021; Wang et al., 2021) for NLP, where the transformations are designed to reveal the brittleness of surface forms, distractions, and prompt-injection faults rather than new semantic meanings of the task. Detailed stress test families are mentioned in Table 3 in detail, and their corresponding method results are shown in Table 4.

D Discovery Prompt Templates and Schemas

This appendix contains all of the automatically generated discovery prompts that were used to develop the frozen LEDSET layers. Only L0 and L1 were derived solely based on the fold-local training split and were both frozen by the time L2 tactical optimization began. I have provided the prompt template(s) as they would need to be replicated for each layer to allow you to replicate the discovery process (exact implementation can be found in the code we release). L0 will initially parse your ground-truth JSON outputs to determine what output keys exist, which is the classification key, which are the label sets, if there are confidence and reasoning fields, etc. Two proposer prompts will then pass a small seeded sample of ground-truth outputs and the first training input through them. The task-realm prompt requires answers to the following schema:

```
{"realm_name":      string,     "realm_-
description":      string,     "realm_-
principles": [string]}
```

Family	Perturbation	Main label-preserving guards
`Disfluency Injection`	Inserts spoken or chat disfluencies, repeats one or two words, or adds a hedge plus trailing filler.	Skips structured inputs and texts shorter than four tokens; preserves the original words.
`Injection (Adversarial)`	Adds adversarial prefix or suffix instructions, including override commands and misleading label metadata.	The injected target is always a non-ground-truth label, prioritized by adjacent-label and semantic-overlap heuristics.
`Irrelevant Context`	Adds unrelated prefix or suffix context while marking the original input as the main request.	Uses distractor focuses from non-ground-truth labels or domain terms; emitted for interactive profiles.
`Keyboard Noise`	Applies light typos, punctuation toggles, doubled spacing, or alternating case.	Inner-letter swaps preserve first letters and recognizability; fallbacks are used only for very short text.
`Mixed Intent`	Adds a secondary request associated with another label to test whether the primary intent remains stable.	Enabled only for interactive datasets with more than two labels; the original request remains present.
`Token Deletion`	Removes one or two mid-utterance content words to test reliance on redundant cues.	Skips structured or very short inputs; preserves the first content word and final token.
`Voice Artifacts`	Simulates speech-to-text artifacts such as lowercasing, punctuation loss, homophones, numeric substitutions, abbreviations, domain-term spellings, and fillers.	Disabled for short, highly structured non-query inputs; excludes substitutions that change pivotal intent verbs.

Table 3: Active metamorphic transformation families used for the reported stress results.

	Ultrasonic NDT							CLINC150						
Method	Disf.	Inj.	Irr.	Key.	Mix.	Tok.	Voi.	Disf.	Inj.	Irr.	Key.	Mix.	Tok.	Voi.
SFT + LEDSET (ours)	**0.94**	**0.91**	**0.94**	**0.94**	**0.92**	**0.94**	0.94	**0.97**	0.93	0.96	0.96	0.85	**0.92**	**0.97**
LEDSET-L0+L1+L2 (ours)	0.92	0.80	0.93	0.92	0.85	0.92	0.93	**0.97**	**0.95**	**0.97**	**0.97**	**0.87**	**0.92**	**0.97**
LEDSET-L2 (ours)	0.91	0.77	0.92	0.92	0.80	0.92	0.93	–	–	–	–	–	–	–
LEDSET-L0+L2 (ours)	0.90	0.83	0.90	0.90	0.83	0.90	0.91	–	–	–	–	–	–	–
GPT-5.1 Baseline	–	–	–	–	–	–	–	0.95	0.94	0.94	0.95	0.72	0.90	0.95
Qwen3:4b LoRA-SFT	**0.94**	0.87	0.91	0.93	0.85	**0.94**	**0.95**	0.96	0.76	0.94	0.96	0.81	**0.92**	**0.97**
EDEO	0.86	0.66	0.85	0.87	0.68	0.88	0.89	0.96	0.85	0.95	0.96	0.67	0.90	0.96
TF-GRPO-C (Cai et al., 2025)	0.78	0.57	0.81	0.79	0.54	0.80	0.79	0.94	0.88	0.93	0.94	0.73	0.89	0.94
APO/ProTeGi (Pryzant et al., 2023)	0.72	0.46	0.71	0.72	0.39	0.72	0.74	0.94	0.78	0.93	0.93	0.80	0.87	0.94
GEPA (Agrawal et al., 2026)	0.80	0.51	0.79	0.79	0.58	0.80	0.81	0.93	0.62	0.93	0.93	0.80	0.88	0.94
MIPROv2 (DSPy, 2026b)	0.74	0.24	0.65	0.73	0.40	0.72	0.72	0.93	0.48	0.92	0.93	0.75	0.88	0.94
COPRO (DSPy, 2026a)	0.61	0.12	0.52	0.60	0.21	0.63	0.61	0.90	0.63	0.87	0.90	0.70	0.85	0.91
Qwen3:4b Baseline (Qwen Team, 2025)	0.58	0.16	0.40	0.58	0.19	0.58	0.58	0.86	0.65	0.84	0.86	0.59	0.82	0.87

Table 4: Best per-transformation accuracy of the methods for the seven metamorphic transformation families on the Ultrasonic NDT and CLINC150 datasets. Columns: Disfluency, Injection, Irrelevant Context, Keyboard Noise, Mixed Intent, Token Deletion, Voice Artifacts.

The persona prompt receives the discovered realm, output structure, and labels, and returns:

```
{"persona": "You are a ... who ..."}
```

The last L0 text concatenates the discovered persona, an output format statement produced by parsing JSON Keys, and the maximum of two discovered realm principles. There are always two fixed axioms that will be added: there must be exactly one label returned for each input, and the classifier cannot create new unsupported information. When the length of the discovered portion exceeds the L0 Character Budget, then the portion can be compressed. The two fixed axioms will be preserved.

For each label y, the discovery process will select up to 20 prototypes (positive) which have the highest ground-truth confidence in addition to selecting up to five negative prototype(s) with high-confidence from every other label. Only the user's query will be displayed on the screen when examples include a dataset wrapper. Any previous conversation context will be summarized if available. The essence extraction prompt will ask the proposer to complete three steps: what is the common intent of all positives? What does this rewritten definition look like using domain neutral architectural concepts? How would you falsify it based upon negatives? The required schema is as follows:

```
{"essence": string, "distinguishing_-
features": [string], "confidence":
number}
```

The prompt explicitly states that trigger words, example specific patterns, thresholds, or domain workflows should not be used because they represent L2 tactics instead of fixed L1 invariant. Once label essences are identified, a Hierarchy Prompt will accept the labels' name and count and deliver semantic clusters and adjacent label pairs:

```
{"clusters":    {string:    [string]},
"adjacent_pairs":    [[string,string]],
"hierarchy_description": string}
```

A second prompt derives three to five disambiguation axioms from the adjacent pairs and hierarchy. It is constrained to design principles rather than surface heuristics, with schema:

```
[{"axiom": string, "rationale": string,
"applies_to": [[string,string]]}]
```

When less than three valid axioms are retrieved, additional default rules will be created as follows in order of preference: first based on intent specificity; secondly for the classification of actions vs. information; and thirdly for classifying based upon the users' primary goal. All of these will then be contained in the compiled L1 block. In addition to the required JSON output contract and at least one compact definition for each label (essence), this block can include up to three ambiguous principle statements. Lastly, there is a single conflict rule that states all L1 definitions have precedence unless they are resolved through an L2 tactic with respect to their definitions.