

Appendix

A Use of LLMs

In this work, LLMs are used strictly for research support rather than as sources of substantive content. Their use falls into: (i) serving as the tested and trained model, and (ii) assisting with language refinement during paper writing. For writing support, we used GPT-5 solely to polish text (improving coherence and grammar) while all ideas, logic, results, and technical contributions originate from the authors.

B Limitations and Future Works

Our study intentionally adopts a controlled closed-set diagnosis setting with immediate feedback, allowing us to isolate and rigorously evaluate the effect of memory-based adaptation independent of parameter updates. This design provides a clean testbed to demonstrate that substantial performance gains can be achieved through experience accumulation and memory updates alone, without modifying model weights. We emphasize that these choices are intentional design decisions to isolate the core research question: *can experience alone, when properly structured and stored, drive meaningful improvement?* By controlling the output space and supervision signal, we create a clean and reproducible environment to rigorously study memory-based adaptation without confounding factors. While real-world clinical environments may involve broader action spaces (e.g., test selection and evidence gathering), and strict safety requirements, we view our current setting as a necessary first step toward such complexity. It establishes a reliable foundation upon which these additional challenges can be systematically incorporated.

C Preliminaries

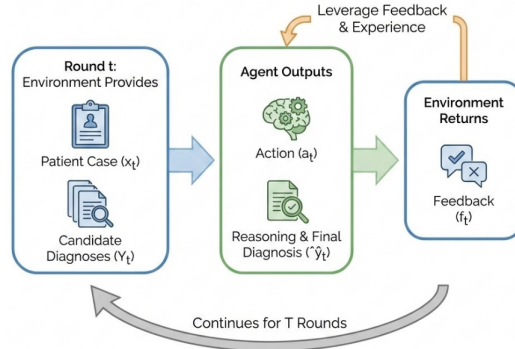


Figure 3: Task setup: At each round t , the environment provides a patient case x_t together with a candidate diagnosis set $\mathcal{Y}_t$. The agent outputs an action a_t consisting of a structured reasoning trace and a final prediction $\hat{y}_t \in \mathcal{Y}_t$. The environment then returns feedback f_t (e.g., correct/incorrect or graded), which the agent can leverage as experience for subsequent rounds. This interaction repeats for T rounds, modeling continual on-the-job improvement in a streaming setting.

Long-Horizon Setting Setup. We study long-horizon disease diagnosis as a sequential decision process motivated by *on-the-job improvement* in deployment. In real clinical settings, a diagnostic agent faces a stream of cases from a specific hospital and population, where prevalence, documentation style, and clinical practice can drift over time. A fixed test-time reasoner may be competent on average yet brittle, repeatedly failing on recurring subtypes or missing subtle local cues. In practice, however, continuously retraining the base model during deployment is often infeasible due to privacy, safety, and operational constraints.

This motivates an agent that can *learn from feedback and experience during use*, improving over the stream via a compact internal state (e.g., memory) rather than frequent model retraining.

At round $t \in \{1, \dots, T\}$, the environment provides a patient case x_t (e.g., demographics, symptoms, and clinical notes) together with a candidate diagnosis set $\mathcal{Y}_t = \{y_t^{(1)}, \dots, y_t^{(m)}\}$. The agent outputs an action a_t that contains (i) a structured reasoning trace and (ii) a final predicted diagnosis $\hat{y}_t \in \mathcal{Y}_t$ (or $\hat{y}_t \in \mathcal{Y}$ in the open-set setting). After acting, the environment returns feedback f_t , which may include correctness, partial credit, or other evaluative signals. The interaction then continues to the next round.

On-the-job improvement objective. Unlike static test-time inference, our objective is continual improvement during deployment: the agent should become more accurate over the stream by incorporating feedback and experience without retraining on the target test stream. Formally, let π_θ denote the policy parameterized by θ . We aim to learn a policy that maximizes the expected cumulative utility over the stream while enabling performance gains over time. Crucially, the agent maintains an internal state s_t across rounds, which summarizes what it has learned so far (e.g., via memory).

Agentic Reinforcement Learning. Following [Zhang et al. \(2025\)](#), we adopt agentic reinforcement learning to solve the task. We optimize the diagnosis agent as a policy model π_θ that generates a structured output o_t (reasoning trace + final prediction) conditioned on the current case and the agent state: $o_t \sim \pi_\theta(\cdot \mid x_t, \mathcal{Y}_t, s_t)$, where the predicted diagnosis $\hat{y}_t$ is parsed from o_t . Each output is scored by a task-specific reward function $r_t = R(x_t, \mathcal{Y}_t, o_t, f_t)$, which may incorporate decision correctness, partial credit, format constraints, and other criteria.

Group Relative Policy Optimization (GRPO). We train the policy with Group Relative Policy Optimization (GRPO) [Shao et al. \(2024\)](#), a PPO-style method that avoids explicit value-function learning by using *within-group* comparisons among multiple sampled outputs for the same input. At each training step for round t , we draw a group of K candidate structured outputs $\{o_t^{(k)}\}_{k=1}^K$ from the current policy $\pi_\theta(\cdot \mid x_t, \mathcal{Y}_t, s_t)$ and score each candidate with our round-wise reward to obtain $\{r_t^{(k)}\}_{k=1}^K$. Instead of learning a critic to predict returns, GRPO estimates an advantage signal by *centering* each candidate reward by the group mean, so candidates that outperform their peers receive positive advantages and those that underperform receive negative advantages. We then update π_θ by increasing the likelihood of higher-advantage outputs while applying (i) a clipped policy-ratio constraint to prevent overly large updates and (ii) a KL regularizer that keeps the updated policy close to a reference policy π_{ref} for stability.

D Details on Candidate Label Selection

A core challenge in constructing candidate diagnosis sets is avoiding *trivial* distractors. If negative options are easily distinguishable, models may succeed by exploiting superficial cues (e.g., matching the affected organ system) rather than performing clinically meaningful discrimination. For example, for a lung-related presentation, the gold diagnosis might involve pulmonary inflammation, while distractors may be unrelated conditions of the leg or heart; such mismatches make the task gameable and overestimate diagnostic ability.

To mitigate this issue, we design distractors to be *clinically plausible* and *semantically close* to the ground-truth label. Rather than sampling negatives uniformly at random from a global label pool (e.g., ~ 800 disease names), we adopt a **cluster-then-sample** strategy: given the gold disease name, we retrieve a set of the most relevant diseases from the label list and sample distractors from this neighborhood. Concretely, we use GPT-5.2 [OpenAI \(2025\)](#) to score candidate disease names by clinical relatedness to the gold label, optionally grouping the label space into coarse clusters (e.g., by organ system, etiology, or phenotype similarity). We then draw N distractors from the top-ranked cluster(s), ensuring that negatives share overlapping symptoms, risk factors, or differential-diagnosis structure with the gold label. This produces candidate sets where success requires fine-grained reasoning (e.g., distinguishing etiologies, time courses, or key discriminative findings) instead of shallow

lexical or organ-level matching. Finally, to ensure fair comparison across methods, we fix the entire candidate-generation pipeline: the distractor pool, retrieval/scoring procedure, distractor count $N = 199$, and random seeds are held constant, so every model is evaluated on identical candidate sets.

E SEA Details

E.1 Dual-Memory Module

We implement the dual-memory module as an external tool, enabling plug-and-play integration with different backbone models without modifying their internal parameters. Specifically, the agent interacts with the memory module through structured tool calls, which support memory writing, retrieval, and consolidation during sequential decision-making.

The dual-memory structure consists of two complementary components: a *short-term memory* that stores recent case-specific experiences, and a *long-term memory* that maintains distilled, reusable knowledge (e.g., diagnostic rules). During inference, the agent retrieves relevant context, appends new experiences after feedback, manages memory, and consolidates abstract rules. We incorporate a system prompt explicitly instructs the agent on when and how to invoke memory operations.

```
{
  "type": "function",
  "function": {
    "name": self.name,
    "description": self.description,
    "parameters": {
      "type": "object",
      "properties": {
        "action": {
          "type": "string",
          "enum": ["list", "append", "pop", "consolidate"],
          "description": (
            "Operation to run. "
            "`list` returns current short-term cases and long
            -term rules. "
            "`append` adds a case record to short-term memory
            . "
            "`pop` evicts cases at given indices from short-
            term memory. "
            "`consolidate` adds distilled diagnostic rules to
            long-term memory."
          ),
        },
        "case_record": {
          "type": "object",
          "description": (
            "Case record to append (used with `append`). "
            "Should contain keys like case_summary, diagnosis
            , feedback, reasoning."
          ),
          "properties": {
            "case_summary": {
              "type": "string",
              "description": "Brief summary of the patient
              case.",
            },
            "diagnosis": {
              "type": "string",
              "description": "The diagnosis that was made."
            }
          }
        }
      }
    }
  }
}
```

```

605         "feedback": {
606             "type": "string",
607             "description": (
608                 "Whether the diagnosis was correct or
609                 incorrect, "
610                 "and the ground truth."
611             ),
612         },
613     },
614 },
615     "indices": {
616         "type": "array",
617         "items": {"type": "integer"},
618         "description": (
619             "Indices of short-term cases to evict (used with
620             `pop`)."
621         ),
622     },
623     "rules": {
624         "type": "array",
625         "items": {"type": "string"},
626         "description": (
627             "Diagnostic rules to add to long-term memory "
628             "(used with `consolidate`)."
629             "Each rule should be a concise, reusable
630             statement "
631             "(e.g. symptom-disease associations)."
632         ),
633     },
634 },
635     "required": ["action"],
636 },
637 },
638 }

```

639 E.2 Training Details

640 **Compute and infrastructure.** Training is conducted on $8 \times$ NVIDIA A100 GPUs. We build
641 our training pipeline on top of the rLLMs infrastructure, which provides distributed rollout
642 sampling, reward computation, and optimizer sharding for efficient policy optimization.
643 The full training consumes about 48 GPU hours for Qwen3-8b base model.

644 **Rollout and truncation.** Each training example corresponds to one round t with patient
645 case x_t and candidates $\mathcal{Y}_t$. We cap the maximum number of generation steps and tool calls
646 per round to 10 turns. A rollout terminates when the model emits a final diagnosis action $\hat{y}_t$.
647 The prompt exceed the max prompt length will be truncate.

648 **Training Parameters.** We adopt Qwen3-4B and Qwen3-8b as the backbone and optimize it
649 with a low learning rate (1e-6) under a PPO framework, using a relatively loose clipping
650 threshold (0.28) and sequence-level loss aggregation. Training is conducted with moderate
651 batch sizes (32 for training) and long context windows (up to 16k tokens for both prompt and
652 response). Rollouts are generated using vLLM in asynchronous mode, with 8 samples per
653 prompt during training (temperature 0.7) and deterministic evaluation settings (1 sample,
654 temperature 0.6, top- p 0.95) for validation. The system runs on a single node with 8 GPUs,
655 incorporating efficiency optimizations such as gradient checkpointing, controlled GPU
656 memory utilization (0.85), and a token limit of 32k per GPU for PPO. Additionally, stepwise
657 advantage estimation is enabled in per-step mode to provide finer-grained credit assignment
658 during training.

659 **Implementation notes.** We keep decoding settings fixed as default across training and
 660 evaluation to reduce confounding factors (temperature, max tokens, and stop conditions),
 661 and we checkpoint the best model based on validation accuracy.

662 F Automatic Evaluation Details

663 F.1 Evaluation Prompts Templates

664 F.1.1 Standard Evaluation Setting

Standard Evaluation Prompt

Role: You are a clinical reasoning evaluator tasked with diagnosing a patient based strictly on provided evidence.

Task:

Given a **Patient Profile** and a list of **Candidate Diseases** (each with descriptions), determine the **single most likely final diagnosis**.

Core Principles:

- This is a **closed-set diagnostic task**: the answer must be selected from the provided Candidate Diseases list.
- Decisions must rely on **explicit evidence** from the Patient Profile.
- Prioritize **clinical consistency, specificity, and minimal contradictions**.

Strict Rules (must follow):

1. Select **EXACTLY ONE** diagnosis from the Candidate Diseases list.
2. Do **NOT** invent diagnoses or use synonyms not appearing in the list.
3. Output the diagnosis name **EXACTLY as written**.
4. Use **ONLY** information from the Patient Profile:
 - Do not assume missing symptoms, labs, or history.
 - Do not rely on external knowledge beyond the provided descriptions.
5. If multiple candidates are plausible, choose the one with:
 - The **most specific and comprehensive match**.
 - The **fewest contradictions**.
6. Prefer diagnoses that explain **key distinguishing features** (e.g., critical symptoms, lab findings, temporal patterns).
7. Do **NOT** output multiple answers, uncertainty, or extra commentary.

Reasoning Requirements:

- Cite **key evidence** from the Patient Profile (symptoms, history, labs, timeline).
- Justify why the selected diagnosis **fits best**.
- Optionally explain why close alternatives are less suitable.
- Keep reasoning **concise, evidence-grounded, and non-speculative**.

Output Format (strict):

Return **ONLY** a valid JSON object (no markdown, no extra text):

```
{
  "reasoning": "Step-by-step evidence-based justification.",
  "final_diagnosis": "EXACT disease name from the candidate list"
}
```

665

Input:**Patient Profile:**

{current_case_prompt}

Candidate Diseases and Descriptions:

{current_choices}

666

Standard Evaluation Prompt Template (ReAct)

Role: You are a clinical reasoning evaluator. You solve diagnosis tasks via iterative reasoning and evidence grounding.

Task:

Given a **Patient Profile** and **Candidate Diseases**, determine the **single most likely final diagnosis**.

Available Information:

- **Patient Profile:** Current case information (symptoms, history, labs).
- **Candidate Diseases:** Closed-set diagnosis options with descriptions.

ReAct Reasoning Process: You must follow an iterative reasoning format:

- **Thought:** Analyze current evidence and identify key clinical signals.
 - **Action:** Examine the patient profile and compare against candidate diseases.
 - **Observation:** Extract useful insights and match findings to candidate descriptions.
- Iteratively repeat the sequence of Thought, Action, and Observation, where each cycle refines your understanding of the case and progressively grounds the diagnosis in evidence.

Then produce a final reasoning summary and diagnosis.

Strict Rules (must follow):

1. Select **EXACTLY ONE** diagnosis from the Candidate Diseases list.
2. Do **NOT** invent diagnoses or use synonyms not in the list.
3. Output the diagnosis name **EXACTLY as written**.
4. Use **ONLY** information from the Patient Profile.
5. Do **NOT** assume missing facts.
6. Choose the diagnosis with:
 - strongest evidence match,
 - highest specificity,
 - minimal contradictions.

Reasoning Requirements:

- Ground decisions in **explicit patient evidence**.
- Prefer concise, structured, and evidence-based reasoning.

Output Format (strict):

Return **ONLY** a valid JSON object (no markdown, no extra text):

```
{
  "reasoning": "Condensed reasoning summary integrating patient evidence",
  "final_diagnosis": "EXACT disease name from the candidate list"
}
```

Input:**Patient Profile:**

{current_case_prompt}

Candidate Diseases and Descriptions:

{current_choices}

667

Standard Evaluation Prompt Template (Memory-Augmented)

Role: You are a clinical reasoning evaluator with access to external memory. You solve diagnosis tasks via iterative reasoning and evidence grounding.

Task:

Given a **Patient Profile**, **Candidate Diseases**, and **Memory**, determine the **single most likely final diagnosis**.

Available Information:

- **Patient Profile:** Current case information (symptoms, history, labs).
- **Candidate Diseases:** Closed-set diagnosis options with descriptions.
- **Memory:** Past cases and history.

ReAct Reasoning Process: You must follow an iterative reasoning format:

- **Thought:** Analyze current evidence and identify key clinical signals.
- **Action:** Decide how to use memory (e.g., retrieve relevant cases or rules).
- **Observation:** Extract useful insights from memory or the patient profile.

Iteratively repeat the sequence of Thought, Action, and Observation, where each cycle refines your understanding of the case and progressively grounds the diagnosis in evidence.

Then produce a final reasoning summary and diagnosis.

Memory Usage Guidelines:

- Use memory as **supporting evidence**, not as a substitute for the Patient Profile.
- Prefer memory entries that match **key attributes** (symptoms, labs, progression).
- **Do NOT override** patient evidence with memory if they conflict.
- Long-term memory (rules) should guide abstraction; short-term memory (cases) should support analogy.

Strict Rules (must follow):

1. Select **EXACTLY ONE** diagnosis from the Candidate Diseases list.
2. Do **NOT** invent diagnoses or use synonyms not in the list.
3. Output the diagnosis name **EXACTLY as written**.
4. Use **ONLY** information from the Patient Profile and provided Memory.
5. Do **NOT** assume missing facts.
6. Choose the diagnosis with:
 - strongest evidence match,
 - highest specificity,
 - minimal contradictions.

Reasoning Requirements:

- Ground decisions in **explicit patient evidence**.
- Incorporate **relevant memory** when helpful.
- Prefer concise, structured, and evidence-based reasoning.

Output Format (strict):

Return **ONLY** a valid JSON object (no markdown, no extra text):

```
{
  "reasoning": "Condensed reasoning summary integrating patient evidence",
  "final_diagnosis": "EXACT disease name from the candidate list"
}
```

Input:
Patient Profile:
 {current_case_prompt}
Candidate Diseases and Descriptions:
 {current_choices}
Memory: {short_term_memory}

669

670 F.1.2 Long-Horizon Evaluation Setting

Long-Horizon Prompt Template

Role: You are a clinical reasoning agent with access to memory, operating in a sequential evaluation setting.

Task: At round t , given the current **Patient Profile**, **Candidate Diseases**, and **Memory**, determine the **single most likely diagnosis**.

Setting:

- Cases arrive sequentially; each case is observed once.
- After each prediction, you receive **feedback** (e.g., correct diagnosis).
- You may update and use **memory** to improve future decisions.

Memory Usage Guidelines:

- Use memory as **supporting evidence**, not a replacement for patient data.
- Do **NOT override** patient evidence with memory if they conflict.

Strict Rules:

1. Select **EXACTLY ONE** diagnosis from the Candidate Diseases.
2. Output the diagnosis name **EXACTLY as written**.
3. Base decisions on Patient Profile, optionally supported by Memory.
4. Do **NOT** assume missing information.

Reasoning Requirements:

- Ground reasoning in explicit patient evidence.
- Incorporate relevant memory when beneficial.
- Clearly connect evidence (and memory, if used) to the diagnosis.

Output Format (strict):

```
{
  "reasoning": "Concise reasoning integrating patient
evidence and relevant memory.",
  "final_diagnosis": "EXACT disease name"
}
```

Memory:

{ $\mathcal{M}$ }

Input (Round t):

Patient Profile:

{ x_t }

Candidate Diseases:

{ $\mathcal{Y}_t$ }

671

Long-Horizon Prompt Template (ReAct)

Role: You are a clinical reasoning agent with access to memory, operating in a sequential evaluation setting. You solve each diagnosis task through iterative reasoning grounded in patient evidence and relevant prior experience.

Task: At round t , given the current **Patient Profile**, **Candidate Diseases**, and **Memory**, determine the **single most likely diagnosis**.

Setting:

- Cases arrive sequentially; each case is observed once.
- After each prediction, you receive **feedback** (e.g., correct diagnosis).
- You may update and use **memory** to improve future decisions.

ReAct Reasoning Process: You must reason in an iterative format:

- **Thought:** Analyze the current case and identify key clinical signals, uncertainties, or distinguishing findings.
- **Action:** Decide whether to inspect the patient profile further, compare candidate diseases, or consult relevant memory.
- **Observation:** Record the useful evidence obtained from the patient profile or memory, and refine the diagnostic hypothesis.

Iteratively repeat the sequence of **Thought**, **Action**, and **Observation** as needed, where each cycle refines your understanding of the case and progressively grounds the diagnosis in evidence.

Memory Usage Guidelines:

- Use memory as **supporting evidence**, not a replacement for patient data.
- Do **NOT** override patient evidence with memory if they conflict.

Strict Rules:

1. Select **EXACTLY ONE** diagnosis from the Candidate Diseases.
2. Output the diagnosis name **EXACTLY as written**.
3. Base decisions on the Patient Profile, optionally supported by Memory.
4. Do **NOT** assume missing information.
5. Do **NOT** invent diagnoses or use synonyms not appearing in the candidate list.

Reasoning Requirements:

- Ground reasoning in explicit patient evidence.
- Incorporate relevant memory only when it provides useful support.
- Clearly connect the selected diagnosis to the strongest supporting evidence.
- Prefer concise, structured, and non-speculative reasoning.

Output Format (strict):

```
{
  "reasoning": "Concise reasoning integrating patient evidence
and relevant memory.",
  "final_diagnosis": "EXACT disease name"
}
```

Memory:

$\{\mathcal{M}\}$

Input (Round t):**Patient Profile:**

$\{x_t\}$

Candidate Diseases:

$\{\mathcal{Y}_t\}$

F.2 Baseline Implementation Details

F.2.1 Standard Evaluation Setting

All baselines share the same input format, candidate set $\mathcal{Y}$, and output requirements (reasoning trace + final prediction), ensuring a controlled comparison.

Zero-shot models directly perform inference using a unified prompting template without any parameter updates. **Supervised Fine-Tuning (SFT)** baselines are trained on the Med-CaseReasoning training split using next-token prediction over reasoning traces and final answers, with the same data preprocessing and candidate sets as SEA. **RL baselines** are optimized with *diagnostic reward only* from raw checkpoint, which provides scalar feedback based solely on final prediction correctness, without intermediate supervision or memory-related rewards. The training configuration (optimizer, batch size, rollout strategy) is kept identical to SEA to isolate the effect of reward design.

For **memory-augmented baselines**, we equip models with the same memory tool interface but remove structured memory management. Specifically, *ReAct+ShortTerm-Memory* uses tool calls to retrieve and append past cases but does not perform consolidation into long-term rules. *SFT+ShortTerm-Memory* follows the same setup but leveraging the SFT-trained model.

F.2.2 Long-Horizon Evaluation Setting

In the streaming setting, all methods operate under a single-pass protocol with frozen model parameters. **Zero-shot** and **trained baselines (SFT, RL)** perform inference independently at each step without retaining past information. **Memory-augmented baselines** extend these models with the dual-memory tool at inference time, allowing them to store case records after feedback and retrieve past experiences for future decisions. However, these baselines do not optimize memory usage.

For **GPT-5.2**, we evaluate both zero-shot and memory-augmented variants under the same protocol. The memory-augmented version uses the identical tool schema and system prompt as SEA, but without any additional training.

G Expert Study Details

We conduct invite professional physician to assess the quality of the updated reasoning and induced rules along four dimensions. Annotator is asked to rate each dimension on a 5-point Likert scale, with higher scores indicating better performance.

Evaluation Criteria.

- **Clinical Correctness.** *Does the updated reasoning and rules remain medically accurate and free of factual errors?*
 - 5: No factual errors; medically sound throughout
 - 4: Very minor factual imprecision with negligible impact
 - 3: Noticeable factual issue that could alter interpretation
 - 2: Major factual error that undermines clinical validity
 - 1: Severely incorrect or unsafe factual content
- **Case Relevance.** *Does the reasoning appropriately incorporate case-specific details?*
 - 5: Fully grounded in key case details and diagnostic context
 - 4: Mostly grounded with minor omissions or weak links
 - 3: Partially grounded; several important details underused
 - 2: Weakly grounded; reasoning is largely generic or misaligned
 - 1: Not grounded in the case; off-target reasoning
- **Rule Usefulness.** *Is the induced rule practically useful and likely to be applied?*

- 719 – 5: Strongly like and would readily apply this rule
- 720 – 4: Generally like and would likely use in most cases
- 721 – 3: Neutral; might use in some situations
- 722 – 2: Dislike; unlikely to use except as a last resort
- 723 – 1: Strongly dislike and would not use this rule
- 724 • **Overall Trust.** *How reliable is the updated reasoning and rule overall?*
- 725 – 5: Very high trust; can be adopted with confidence
- 726 – 4: High trust; acceptable with minor caveats
- 727 – 3: Moderate trust; requires targeted review
- 728 – 2: Low trust; substantial concerns remain
- 729 – 1: Very low trust; should not be adopted

730 **Annotation Guidelines.** Annotator is instructed to consider both the updated reasoning
731 trace and the induced rules jointly. Clinical correctness should be judged strictly based on
732 medical validity, while case relevance emphasizes grounding in the specific patient context.
733 Rule usefulness captures practical applicability, and overall trust reflects a holistic judgment
734 combining all aspects.

735 **Annotator Profile.** Annotator is an internal medicine resident with a strong interdisciplinary
736 background in clinical medicine, biostatistics, and oncology research. Recipient of multiple
737 academic distinctions, including national and institutional scholarships.

738 H Case Study

739 We present 15 trajectory from **SEA** under the long-horizon setting in the supplementary
740 materials to illustrate how the agent adapts its reasoning and memory over time to resolve
741 complex diagnostic scenarios.