ORACLERESOLVE: HYPOTHESIS-GUIDED EVIDENCE REASONING FOR ORACLE BONE SCRIPT DECIPHERMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Oracle bone script (OBS) decipherment requires reasoning across heterogeneous evidence, including glyph morphology, inscriptional context, diachronic character forms, and philological scholarship. Existing computational approaches increasingly incorporate such evidence, yet fixed evidence pipelines and generic tool-routing agents do not explicitly couple competing interpretations with the selection of subsequent evidence. This limitation is particularly important in OBS decipherment, where multiple morphologically plausible readings may require different contextual, historical, or philological evidence to distinguish, and where the correct interpretation may be absent from the initial candidate set. We introduce **OracleResolve**, a hypothesis-guided research agent that maintains candidate interpretations as persistent hypotheses together with supporting evidence, contradicting evidence, scores, and unresolved questions. Morphology-based Glyph and Component Experts first construct the initial hypothesis set from the target glyph. During subsequent reasoning, OracleResolve identifies unresolved disagreements among the current hypotheses and selectively queries contextual, diachronic, or philological evidence. Retrieved evidence is explicitly evaluated to revise the hypothesis state and subsequent investigation; when the accumulated evidence cannot adequately support the current candidate set, an open-candidate retrieval module expands the hypothesis space with new interpretations. We evaluate OracleResolve on a class-ID-disjoint OBS decipherment setting with an additional stricter reading-unseen subset, while preventing evaluation source groups from entering the contextual retrieval corpus. Controlled comparisons against direct multimodal reasoning, a fixed evidence pipeline, and a generic tool-routing agent show that OracleResolve improves decipherment reliability, better recovers from incorrect initial preferences and incomplete hypothesis spaces, without increasing average evidence-tool usage. These results support hypothesis-guided evidence reasoning as a promising framework for computational ancient-script decipherment.

1 INTRODUCTION

Oracle bone script (OBS) decipherment aims to develop testable interpretations of a character's identity and its usage in inscriptional contexts from fragmentary glyph forms and related historical evidence. Unlike closed-set glyph recognition, decipherment requires reasoning across multiple sources of evidence, including component structure, historical variants, inscriptional occurrences, and previous philological scholarship. Visual similarity can provide plausible candidate readings, but a credible interpretation must also remain compatible with contextual, diachronic, and scholarly evidence. Computational decipherment should therefore go beyond producing a single plausible answer and instead support interpretations that can be examined, compared, and revised as additional evidence becomes available.

Recent work has expanded computational OBS research beyond isolated visual recognition (Hu et al., 2025; Li et al., 2023) toward knowledge-enhanced understanding and evidence-supported decipherment. OBI-Bench (Chen et al., 2025) systematically evaluates large multimodal models across recognition, rejoining, classification, retrieval, and decipherment tasks, demonstrating both the potential and the remaining limitations of general-purpose multimodal models for OBS research. OBB

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

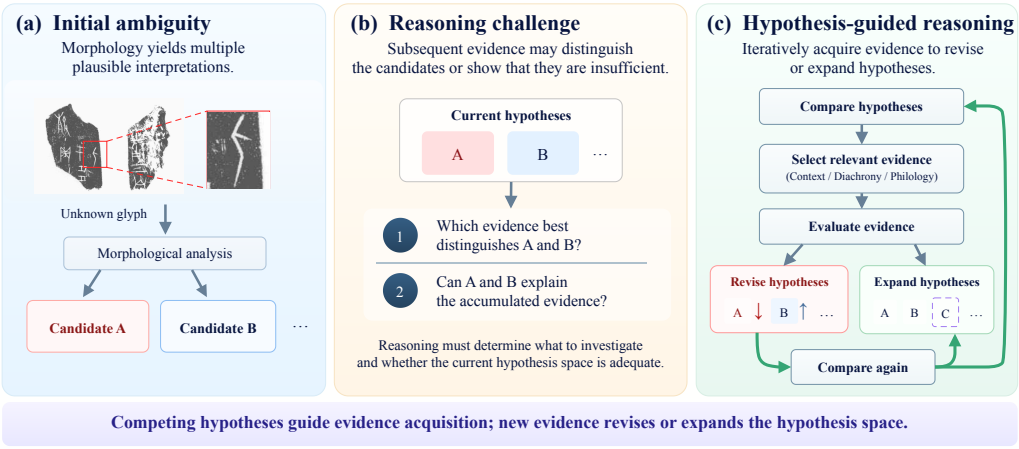


Figure 1: **Why hypothesis-guided evidence reasoning is needed.** Initial morphological analysis may yield multiple plausible interpretations of an unknown oracle bone glyph. OracleResolve uses unresolved disagreements among these hypotheses to guide subsequent contextual, diachronic, and philological evidence acquisition; new evidence can either revise existing hypotheses or expand the hypothesis space when the current candidates are insufficient.

and OBS-PRTF (Liu, 2027) further emphasize structured, evidence-grounded reasoning that can be inspected by domain experts. Component-grounded approaches (Zhang et al., 2026) combine visual grounding with structured knowledge retrieval and relation reasoning to exploit reusable semantic components. At the system level, AlphaOracle (Liu et al., 2026) integrates morphological analysis, contextual evidence, diachronic forms, and philological literature following an expert-inspired workflow, whereas OracleAgent (Li et al., 2025a) uses an LLM controller to flexibly orchestrate specialized OBS tools and multimodal knowledge resources. Together, these developments substantially broaden the evidence available to computational decipherment (Chen et al., 2026).

Recent agentic reasoning methods enable models to interact with external tools and to preserve alternative reasoning paths, yet these capabilities do not by themselves specify how conflicting domain interpretations should govern the acquisition of heterogeneous evidence (Yao et al., 2023b;a; Wang et al., 2026; Schick et al., 2023).

However, OBS decipherment poses a distinctive challenge for hypothesis-driven reasoning. Candidate interpretations must be evaluated against heterogeneous evidence, including glyph morphology, inscriptional contexts, diachronic character forms, and philological scholarship (Qiao et al., 2025). Different evidence sources may favor different interpretations: a candidate can be visually plausible yet inconsistent with contextual usage, while another may fit the inscriptional context but lack diachronic support. Moreover, the initial candidate set itself may be incomplete, especially for previously unseen or disputed characters. Therefore, maintaining and reranking multiple candidates alone is insufficient. A decipherment system must determine which unresolved disagreement should be investigated next, revise existing hypotheses as new evidence arrives, and reopen the hypothesis space when the accumulated evidence cannot be adequately explained by any current candidate.

Consider two candidate readings that are both compatible with a local glyph component but imply different contextual usages. Retrieving additional visually similar glyphs may provide little information for distinguishing them, whereas examining inscriptional contexts that differentiate their predicted usages may directly resolve the disagreement (Asai et al., 2024). Conversely, when two candidates are equally compatible with the available contexts but imply different historical correspondences, diachronic evidence may be more informative. These observations motivate *hypothesis-guided evidence reasoning*, where disagreements among persistent candidate interpretations guide the selection of subsequent evidence, and newly acquired evidence in turn revises the hypotheses and determines the next investigation step. More fundamentally, if newly acquired evidence is inconsistent with all current candidates, the appropriate response is not to force a choice among them, but to reopen candidate generation and search for an alternative interpretation. Figure 1 illustrates

this motivation: competing hypotheses guide which evidence should be acquired next, while newly acquired evidence can either revise the current ranking or reveal that the hypothesis space itself is incomplete.

To instantiate this principle, we introduce **OracleResolve**, an agentic framework that couples persistent hypothesis maintenance with disagreement-driven evidence acquisition and hypothesis expansion. OracleResolve maintains multiple candidate interpretations together with the evidence that supports or challenges them and the questions that remain unresolved. At each reasoning step, the agent assesses whether the current hypothesis set should be further investigated, expanded, or terminated. When plausible alternatives remain unresolved, OracleResolve identifies a disagreement that could change their relative plausibility and selects the most relevant contextual, diachronic, or philological evidence source. The retrieved evidence is then used to revise the hypothesis state and guide the next investigation step. When the accumulated evidence cannot be adequately explained by any current candidate, OracleResolve reopens hypothesis generation instead of forcing a decision within an incomplete candidate set. This closed loop design enables both recovery from an incorrect initial ranking and recovery from an incomplete initial hypothesis space.

We evaluate OracleResolve under controlled settings with matched backbone models, evidence resources, tool interfaces, and evidence budgets. Compared with direct multimodal reasoning, fixed evidence pipelines, and generic tool-routing agents, OracleResolve improves Top-1 decipherment accuracy by **4.10** points. It recovers **60.00%** of cases with an incorrect initial top-ranked hypothesis, while no erroneous revisions were observed among initially correct predictions.

In summary, our contributions are threefold:

- We introduce **OracleResolve**, an agentic framework for OBS decipherment that maintains competing interpretations as persistent hypotheses and couples them with iterative evidence acquisition and revision.
- We develop a disagreement-driven reasoning mechanism in which unresolved differences among candidate interpretations guide heterogeneous evidence acquisition, while newly acquired evidence supports hypothesis revision and triggers hypothesis expansion when the current candidate set becomes insufficient.
- We establish a controlled evaluation setting that isolates the contribution of hypothesis-guided reasoning from backbone models and evidence resources, and show that OracleResolve improves decipherment accuracy, recovery from incorrect initial preferences, and evidence efficiency.

2 Related Work

Oracle Bone Script Understanding and Decipherment. Computational research on oracle bone script (OBS) has gradually expanded from isolated recognition and retrieval tasks toward knowledge-intensive interpretation and decipherment. OBI-Bench (Chen et al., 2025) provides a holistic evaluation of large multimodal models across recognition, rejoining, classification, retrieval, and decipherment, revealing substantial challenges for general-purpose models in tasks requiring fine-grained visual perception and OBS-specific knowledge. More recent approaches incorporate structured domain knowledge and explicit evidence into the reasoning process. Component-grounded methods (Zhang et al., 2026) combine visual grounding with knowledge retrieval and relation reasoning to exploit recurrent pictographic components, while OBB/OBS-PRTF (Liu, 2027) emphasizes structured and evidence-grounded OBS understanding. At the system level, AlphaOracle (Liu et al., 2026) follows an expert-inspired workflow that progressively integrates morphological, contextual, diachronic, and philological evidence, whereas OracleAgent (Li et al., 2025a) uses an LLM controller to orchestrate specialized OBS tools and multimodal knowledge resources. These works substantially broaden the evidence available to computational decipherment (Li et al., 2025b). Our work builds on this progress but focuses on a different question: rather than following a predefined evidence workflow or routing tools primarily according to the current task, we investigate how disagreements among persistent candidate interpretations can directly guide subsequent evidence acquisition and revision.

Agentic Reasoning and Hypothesis Verification. LLM-based agents extend language models with external actions, observations, and iterative reasoning. ReAct (Yao et al., 2023b) interleaves reasoning and tool interaction, while Tree of Thoughts (Yao et al., 2023a) demonstrates the benefit of maintaining and evaluating alternative reasoning trajectories. Recent work has further explored explicit hypothesis verification. VideoHV-Agent (Wang et al., 2026), for example, reformulates long-video question answering by converting candidate answers into testable hypotheses, deriving discriminative clues, and verifying them against localized video evidence. This work is closely related to our motivation, but differs in both the hypothesis and evidence spaces: its hypotheses are derived from a provided answer set and are verified primarily against a common video source, whereas OBS decipherment requires generating and revising candidate interpretations using heterogeneous evidence from glyph morphology, inscriptional contexts, diachronic forms, and philological literature (Shinn et al., 2023). OracleResolve therefore studies hypothesis-guided reasoning in a knowledge-intensive, open-ended setting, where a single research agent must determine not only what evidence could distinguish competing interpretations, but also which specialized evidence source should be consulted and how newly acquired evidence should revise the subsequent investigation.

3 ORACLERESOLVE

3.1 PROBLEM SETUP AND FRAMEWORK OVERVIEW

OracleResolve contains three functionally distinct classes of modules: proposal experts for constructing the initial hypothesis space, evidence tools for testing competing hypotheses, and an open-candidate generator for expanding the hypothesis space when existing candidates become insufficient. A single central LLM controller coordinates these modules throughout the reasoning process, and the framework is agnostic to the specific backbone used for the controller.

Given a target oracle bone glyph g and, when available, its local inscriptional context c, our goal is to infer plausible interpretations while preserving the evidence that supports or challenges competing alternatives. Unlike one-shot glyph recognition or candidate reranking over a fixed answer set, we formulate decipherment as an iterative and open-ended evidence-guided reasoning process. Starting from an initial hypothesis set $\mathcal{H}_0$, OracleResolve repeatedly acquires external evidence, revises the relative plausibility of existing hypotheses, and determines what should be investigated next. Importantly, the candidate space itself is not assumed to be complete: when the accumulated evidence cannot be adequately explained by any current hypothesis, the system reopens hypothesis generation and introduces new candidate interpretations. The final output is therefore not merely the best candidate from a fixed initial set, but a ranked set of hypotheses obtained through iterative evidence acquisition, revision, and, when necessary, hypothesis expansion.

At reasoning step t, OracleResolve maintains a dynamic hypothesis set

$$\mathcal{H}_t = \{H_i^t\}_{i=1}^{K_t},$$

where K_t denotes the number of active candidate hypotheses at step t. Each hypothesis is represented as

$$H_i^t = \left(y_i, s_i^t, \mathcal{E}_{i,+}^t, \mathcal{E}_{i,-}^t, Q_i^t\right),$$

where y_i denotes the candidate interpretation, s_i^t its current score, $\mathcal{E}_{i,+}^t$ and $\mathcal{E}_{i,-}^t$ denote the supporting and contradicting evidence accumulated for the candidate, respectively, and Q_i^t records unresolved questions that may distinguish it from competing interpretations. We additionally maintain a global evidence state $\mathcal{E}_t$, which stores all evidence acquired up to step t together with its provenance. This representation allows the agent to reason not only about which candidate is currently preferred, but also about why it is preferred, which evidence challenges it, and what remains unexplained by the current hypothesis set.

OracleResolve separates *initial hypothesis proposal* from *subsequent evidence acquisition*. Given the target glyph g, two complementary perception experts first construct the initial candidate space. A glyph expert P_{glyph} focuses on global shape and visual similarity, whereas a component expert P_{comp} analyzes component structure and compositional cues. Their outputs are combined into shared initial evidence $\mathcal{E}_0$ and used to initialize the candidate hypothesis set $\mathcal{H}_0$. These proposal

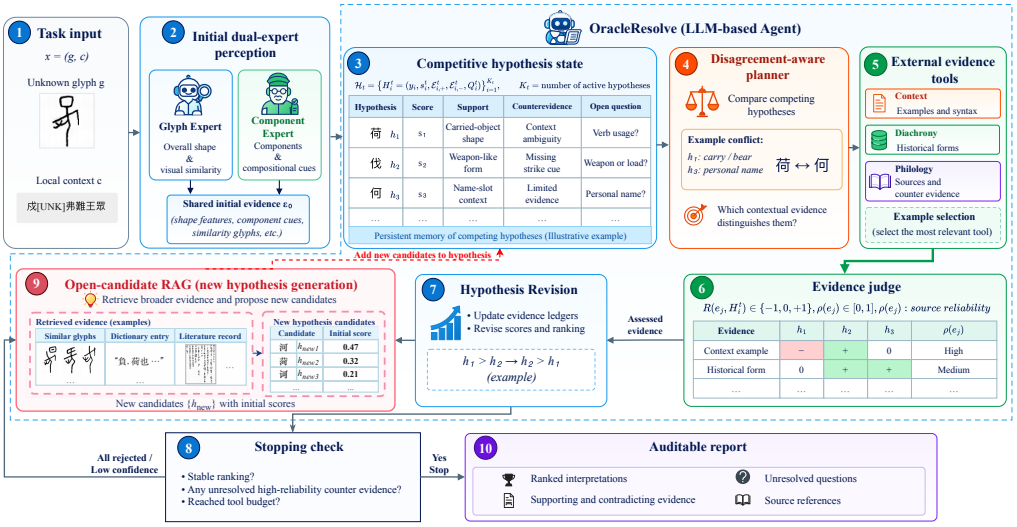


Figure 2: **Overview of OracleResolve.** Glyph and Component Experts first initialize morphology-based candidate hypotheses. OracleResolve then maintains these candidates as persistent competing hypotheses and uses unresolved disagreements to guide the acquisition of contextual, diachronic, and philological evidence. Retrieved evidence is judged to revise the hypothesis state and subsequent investigation. When the current candidate set becomes insufficient, open-candidate RAG expands the hypothesis space; otherwise, the reasoning loop continues until termination and produces an auditable, evidence-grounded report.

experts are applied only during hypothesis initialization and are shared across all compared reasoning strategies, ensuring that subsequent differences do not arise from stronger initial candidate generation.

After initialization, OracleResolve reasons over a heterogeneous set of external evidence tools $\mathcal{T} = \{T_{\text{ctx}}, T_{\text{dia}}, T_{\text{phil}}\}$. The contextual tool T_{ctx} retrieves inscriptional occurrences and surrounding contexts relevant to candidate usage; the diachronic tool T_{dia} provides related character forms across historical stages for examining possible evolutionary correspondences; and the philological tool T_{phil} retrieves dictionary entries and scholarly discussions associated with the current interpretations. Unlike the initialization experts, these tools are selected dynamically according to the unresolved disagreement among the current hypotheses. Their outputs provide evidence rather than final decisions and are subsequently evaluated by the evidence judge. The same evidence tools are made available to all compared reasoning strategies in our controlled experiments.

When accumulated evidence indicates that the current candidate set is insufficient, OracleResolve invokes an open-candidate hypothesis expansion module G_{exp}. Rather than merely reranking existing interpretations, G_{exp} retrieves broader morphological and scholarly evidence and proposes additional candidate hypotheses, which are then inserted into the persistent hypothesis state and evaluated through the same reasoning loop.

Figure 2 and Algorithm 1 summarize the overall loop. After morphology-based initialization, the controller repeatedly chooses among evidence investigation, hypothesis expansion, and termination; Sec. 3.2 details these operations.

3.2 Hypothesis-Guided Evidence Reasoning

Hypothesis state. The hypothesis state defined in Sec. 3.1 serves as the persistent memory of OracleResolve. In particular, supporting and contradicting evidence distinguish hypotheses weakened by explicit counterevidence from those that remain uncertain due to insufficient investigation, while the unresolved questions Q_i^t expose what still needs to be tested. The disagreement analyzer operates over these states to identify candidate distinctions that may change the current ranking. Initial morphological evidence produced by the glyph and component experts remains part of the

Algorithm 1 OracleResolve

Require: Target glyph g, optional context c, evidence tools $\mathcal{T}$, reasoning budget B

$(\mathcal{H}_0, \mathcal{E}_0) \leftarrow \textsc{InitialPerception}(g)$
for $t = 0, \ldots, B-1$ **do**
  $m_t \leftarrow \textsc{AssessState}(\mathcal{H}_t, \mathcal{E}_t)$
  if $m_t = \textsc{Stop}$ **then**
    break
  else if $m_t = \textsc{Expand}$ **then**
    $(\Delta\mathcal{H}_t, \Delta\mathcal{E}_t^{\text{exp}}) \leftarrow \textsc{ExpandHypotheses}(g, c, \mathcal{H}_t, \mathcal{E}_t)$
    $(\mathcal{H}_{t+1}, \mathcal{E}_{t+1}) \leftarrow \textsc{MergeExpansion}(\mathcal{H}_t, \mathcal{E}_t, \Delta\mathcal{H}_t, \Delta\mathcal{E}_t^{\text{exp}})$
  else
    $\delta_t \leftarrow \textsc{IdentifyDisagreement}(\mathcal{H}_t, \mathcal{E}_t)$
    $(a_t, q_t) \leftarrow \textsc{SelectEvidence}(\mathcal{H}_t, \mathcal{E}_t, \delta_t, c, \mathcal{T})$
    $\Delta\mathcal{E}_t \leftarrow T_{a_t}(q_t)$
    $(\mathcal{H}_{t+1}, \mathcal{E}_{t+1}) \leftarrow \textsc{Revise}(\mathcal{H}_t, \mathcal{E}_t, \Delta\mathcal{E}_t)$
  end if
end for
return ranked hypotheses with supporting and contradicting evidence

evidence state throughout reasoning, although the experts themselves are not dynamically invoked after initialization.

Disagreement-driven evidence acquisition. At each reasoning step, OracleResolve first identifies an unresolved disagreement among the active hypotheses. We formalize this operation as $\delta_t = D(\mathcal{H}_t, \mathcal{E}_t)$, where δ_t specifies both the competing hypotheses involved and the unresolved issue that may change their relative plausibility. The planner then selects an evidence source and formulates a targeted query,

$$(a_t, q_t) = \pi(\mathcal{H}_t, \mathcal{E}_t, \delta_t, c), \qquad a_t \in \{T_{\text{ctx}}, T_{\text{dia}}, T_{\text{phil}}\}.$$

The optional context c is not used to construct $\mathcal{H}_0$, but becomes available during subsequent evidence reasoning. T_{ctx} retrieves inscriptional examples relevant to the disputed usage, T_{dia} retrieves historical forms that can distinguish competing diachronic interpretations, and T_{phil} retrieves dictionary entries and scholarly discussions relevant to unresolved philological claims. Importantly, q_t is formulated to discriminate among competing hypotheses rather than merely retrieve additional support for the currently top-ranked interpretation.

Evidence-grounded hypothesis revision. Executing the selected evidence tool returns a set of newly retrieved evidence, $\Delta\mathcal{E}_t = T_{a_t}(q_t)$. Each evidence item $e_j \in \Delta\mathcal{E}_t$ is accompanied by provenance information and evaluated against the relevant hypotheses by the evidence judge,

$$R(e_j, H_i^t) \in \{-1, 0, +1\},$$

where $+1$, -1, and 0 denote supporting, contradicting, and non-discriminative evidence, respectively. Each evidence item is additionally assigned a source-reliability weight $\rho(e_j) \in [0, 1]$. The hypothesis score is updated as

$$s_i^{t+1} = s_i^t + \eta \sum_{e_j \in \Delta\mathcal{E}_t} \rho(e_j) R(e_j, H_i^t),$$

where η controls the contribution of newly acquired evidence. The revision operator U jointly updates the scores, supporting and contradicting evidence ledgers, unresolved questions, and the global evidence state:

$$(\mathcal{H}_{t+1}, \mathcal{E}_{t+1}) = U(\mathcal{H}_t, \mathcal{E}_t, \Delta\mathcal{E}_t).$$

In particular, $\mathcal{E}_{t+1} = \mathcal{E}_t \cup \Delta\mathcal{E}_t$, while evidence classified as supporting or contradicting is appended to the corresponding hypothesis-specific ledger. Resolved questions are removed from Q_i^t, and newly exposed uncertainties may be added. Thus, each ranking change remains traceable to the evidence that caused it.

Replanning, hypothesis expansion, and termination. After each revision, OracleResolve reassesses the current state and chooses among three actions: INVESTIGATE, EXPAND, or STOP. If plausible hypotheses remain but important disagreements are unresolved, the agent continues the evidence-acquisition loop. If the current hypothesis set becomes inadequate. For example, when no active candidate can sufficiently account for the accumulated evidence or all candidates are substantially challenged by reliable counterevidence. OracleResolve invokes the open-candidate generator G_{exp}. Unlike ordinary evidence acquisition, expansion explicitly reopens the candidate space by retrieving broader morphological and scholarly material and proposing interpretations that are absent from the current hypothesis set:

$$(\Delta\mathcal{H}_t, \Delta\mathcal{E}_t^{\text{exp}}) = G_{\text{exp}}(g, c, \mathcal{H}_t, \mathcal{E}_t).$$

Here, $\Delta\mathcal{E}_t^{\text{exp}}$ denotes the additional evidence retrieved during open-candidate search. The proposed candidates are initialized as new hypothesis states and their associated evidence is evaluated by the same evidence judge used during ordinary revision, rather than being treated as automatically supporting the newly generated candidates. After duplicate removal and evidence evaluation, the new hypotheses and expansion evidence are merged into the persistent state:

$$\mathcal{H}_{t+1} = \text{Merge}(\mathcal{H}_t, \Delta\mathcal{H}_t), \qquad \mathcal{E}_{t+1} = \mathcal{E}_t \cup \Delta\mathcal{E}_t^{\text{exp}}.$$

The newly introduced hypotheses then participate in subsequent disagreement identification and evidence acquisition under the same reasoning protocol. This mechanism enables OracleResolve to recover both from an incorrect ranking within the initial candidate set and from an incomplete initial hypothesis space. The process terminates when the current state satisfies the predefined stopping criteria or when the reasoning budget B is exhausted. If sufficient evidence is still unavailable at termination, OracleResolve retains the unresolved candidates rather than forcing a definitive interpretation.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Evaluation protocol. We evaluate on 44 held-out oracle-bone character classes, with two instances per class and 88 test cases in total. The proposal experts are trained on 1,028 class IDs, while all evaluation classes are drawn from a disjoint 734-class candidate pool. Each test instance contains a target glyph crop and an inscriptional context with the target position replaced by `[UNK]`; private reference readings are used only for offline scoring. For every test class, the two evaluation instances come from source groups that are globally excluded from the visible context corpus, preventing the same inscriptional material from appearing in both evaluation and retrieval. Because 20 test instances share an accepted reading string with a training-class name despite class-ID disjointness, we report results on both the full 88-instance set and a stricter 68-instance reading-unseen subset. Dataset construction, filtering criteria, and source-group statistics are detailed in Appendix A.

Baselines. We compare DIRECT-LLM, FIXED-PIPELINE, GENERIC-AGENT, and ORACLERESOLVE. DIRECT-LLM predicts directly from the target image and context without local experts or evidence tools. The remaining methods share the same initial hypothesis set, proposal experts, evidence resources, and reasoning budget. FIXED-PIPELINE queries contextual, diachronic, and philological evidence in a predefined order; GENERIC-AGENT dynamically selects tools without OracleResolve's explicit competing-hypothesis state and score-based revision; and ORACLERESOLVE additionally uses disagreement-driven acquisition, supporting/contradicting evidence memory, iterative revision, replanning, and hypothesis expansion.

Metrics. We report Top-1 accuracy, Top-k accuracy, and mean reciprocal rank with respect to each instance's accepted reference-reading set. To evaluate revision, we separately measure recovery when the initial Top-1 hypothesis is incorrect but a correct reading is already present in $\mathcal{H}_0$, together with the erroneous-revision rate on initially correct cases. To evaluate hypothesis expansion, we report recovery on cases where no accepted reading is present in $\mathcal{H}_0$. Evidence efficiency is measured by the average number of tool calls per instance.

Table 1: Overall decipherment performance on the full class-ID-disjoint test set and the stricter reading-unseen subset. All matched methods use the same initial hypotheses, evidence resources, and reasoning budget.

Method	Full Set			Reading-Unseen			Calls
	Top-1 ↑	Top-3 ↑	MRR ↑	Top-1 ↑	Top-3 ↑	MRR ↑	Avg. ↓
Direct-LLM	1.14	4.55	3.41	1.47	1.47	1.84	–
Fixed-Pipeline	11.90	28.57	19.61	6.25	18.75	11.68	3.00
Generic-Agent	22.73	29.55	25.76	19.12	26.47	22.30	2.69
OracleResolve	**26.83**	**37.80**	**33.21**	**22.58**	**30.65**	**26.85**	**2.67**

Table 2: Hypothesis recovery under incorrect and incomplete initial hypothesis states. All matched reasoning methods have access to the same initial hypotheses, evidence resources, open-candidate generator, and reasoning budget. Rank Recovery evaluates cases where an accepted reading is present in $\mathcal{H}_0$ but is not initially ranked first; Erroneous Revision measures incorrect changes to initially correct Top-1 predictions. Expansion Recall and Expansion Top-1 evaluate recovery when no accepted reading is present in $\mathcal{H}_0$.

Method	Rank Rec. ↑	Err. Rev. ↓	Exp. Rec. ↑	Exp. Top-1 ↑
Fixed-Pipeline	0.00	0.00	19.12	4.41
Generic-Agent	11.11	14.29	25.93	19.75
OracleResolve	**60.00**	**0.00**	**31.82**	**22.73**

Implementation. All agent-based methods use the same central model identifier `gpt-6-sol` and matched reasoning configurations. Each instance is limited to at most three reasoning rounds and three evidence-tool calls; initial hypothesis generation is excluded from this controller budget. The Glyph and Component Experts construct the shared initial hypothesis set exclusively from the target image, while contextual information is introduced only during subsequent evidence reasoning. Detailed model architectures, retrieval implementations, evidence-reliability settings, decision thresholds, API configurations, and hyperparameters are provided in Appendix E.

4.2 Main Results

Table 1 summarizes the overall decipherment performance. OracleResolve achieves the best Top-1 accuracy, Top-k accuracy, and MRR on both the full evaluation set and the stricter reading-unseen subset, improving Top-1 accuracy by 4.10 percentage points on the full set and by 3.46 points on the stricter reading-unseen subset over the strongest baseline. The comparison between Fixed-Pipeline and Generic-Agent measures the benefit of adaptive tool use, while the additional gain of OracleResolve over Generic-Agent isolates the effect of hypothesis-guided evidence acquisition and revision under matched tools and budgets.

OracleResolve also requires 2.67 evidence calls on average, compared with 2.69 for the generic agent and 3.00 for the fixed pipeline. This suggests that OracleResolve achieves these gains without increasing the average number of evidence calls.

4.3 Hypothesis Recovery and Ablation

Recovery from incorrect initial preferences. We first isolate cases where an accepted reading is already present in $\mathcal{H}_0$ but is not initially ranked first. OracleResolve recovers 60.00% of these cases, substantially higher than the generic agent 11.11% and the fixed pipeline 0.00%. Importantly, OracleResolve introduces no erroneous revisions of initially correct predictions, whereas the generic agent changes 14.29% of initially correct predictions to incorrect ones. This suggests that the gain in recovery does not result from indiscriminately revising initial predictions, but from more selective evidence-guided hypothesis revision.

Table 3: Ablation study of OracleResolve on the full class-ID-disjoint evaluation set. All variants use the same backbone, initial hypotheses, evidence resources, open-candidate generator, with only the specified reasoning component removed or replaced. Δ Top-1 denotes the change relative to the full OracleResolve model, with negative values indicating performance degradation.

Variant	Top-1 ↑	MRR ↑	Exp. Rec. ↑	Δ Top-1
OracleResolve	**26.83**	**33.21**	**31.82**	–
w/o Multi-Hypothesis	23.86	23.86	19.44	-2.97
w/o Contradiction Memory	11.36	17.82	15.28	-15.47
w/o Replanning	10.23	18.67	22.22	-16.60
w/o Disagreement-Driven Acquisition	9.09	16.94	19.44	-17.74
w/o Hypothesis Expansion	7.95	11.40	–	-18.88

Recovery from incomplete hypothesis spaces. We next consider cases where no accepted reference reading appears in the shared initial hypothesis set. All matched reasoning methods are given access to the same open-candidate generator, allowing us to evaluate how different reasoning strategies use hypothesis expansion rather than whether expansion is available at all. OracleResolve achieves the highest Expansion Recall at 31.82%, compared with 25.93% for the generic agent and 19.12% for the fixed pipeline. It also obtains the highest Expansion Top-1 rate at 22.73%, compared with 19.75% and 4.41%, respectively. These results indicate that access to open-candidate retrieval alone is insufficient: the way newly introduced hypotheses are triggered, evaluated, and incorporated into subsequent reasoning also affects recovery from an incomplete initial hypothesis space.

Ablation analysis. Table 3 examines the contribution of individual components while keeping the backbone, initial hypotheses, evidence resources, and reasoning budget fixed. Every ablation reduces Top-1 accuracy, with drops ranging from 2.97 to 18.88 percentage points. The largest degradation occurs when hypothesis expansion is disabled, reducing Top-1 accuracy from 26.83% to 7.95% (-18.88 points). Replacing disagreement-driven acquisition with a fixed evidence order also causes a substantial drop (-17.74 points), followed by disabling replanning (-16.60 points) and removing contradiction memory (-15.47 points). In contrast, removing multi-hypothesis reasoning produces a smaller Top-1 decrease (-2.97 points), although it also weakens recovery from incomplete initial hypothesis spaces. These results indicate that OracleResolve benefits from both open-candidate recovery and the iterative mechanisms used to acquire, retain, and reassess evidence.

4.4 Qualitative Analysis

OracleResolve can recover both from an incorrect preference within the current hypothesis set and from an incomplete hypothesis space. A representative end-to-end trajectory illustrating closed-set revision, hypothesis expansion, and subsequent evidence-based reranking is provided in Appendix B.

5 Conclusion

We introduced OracleResolve, a hypothesis-guided reasoning framework for oracle bone script decipherment that treats candidate interpretations as persistent competing hypotheses rather than transient predictions. By coupling disagreement-driven evidence acquisition with explicit supporting and contradicting evidence, iterative hypothesis revision, and open-candidate expansion, OracleResolve can address both incorrect preferences within the current candidate set and cases in which the initial hypothesis space is incomplete. Controlled experiments under matched backbone models, evidence resources, and reasoning budgets show consistent improvements over fixed and generic agent baselines on both the full class-ID-disjoint evaluation set and the stricter reading-unseen subset, while the recovery and ablation analyses further demonstrate the importance of adaptive evidence acquisition and hypothesis expansion. Future work will extend OracleResolve to larger corpora, raw rubbing inputs, and expert-validated decipherment settings.

AI USE STATEMENT

Generative AI tools were used to assist with literature discovery, drafting and language editing, and code-related assistance. All methodological decisions, implementations, experimental results, citations, and claims were reviewed and verified by the authors. The authors take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

We provide the complete dataset construction and leakage-control protocol, model and retrieval configurations, evidence-scoring rules, decision thresholds, prompts, and additional experimental details in the appendix. All controlled baselines share the same proposal experts, evidence resources, and reasoning budgets unless otherwise specified.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.

Zijian Chen, tingzhu chen, Wenjun Zhang, and Guangtao Zhai. Obi-bench: Can lmms aid in study of ancient script on oracle bones? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 102851–102881, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ff7bf6014f7826da531aa50f4538ee19-Paper-Conference.pdf.

Zijian Chen, Wenjie Hua, Jinhao Li, Lirong Deng, Fan Du, Tingzhu Chen, and Guangtao Zhai. Pictobi-20k: Unveiling large multimodal models in visual decipherment for pictographic oracle bone characters. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11937–11941, 2026. doi: 10.1109/ICASSP55912.2026.11462601.

Haisu Guan, Jinpeng Wan, Yuliang Liu, Pengjie Wang, Kaile Zhang, Zhebin Kuang, Xinyu Wang, Xiang Bai, and Lianwen Jin. An open dataset for the evolution of oracle bone characters: Evobc, 2024. URL https://arxiv.org/abs/2401.12467.

Jingcheng He. *Jiaguwen Zi Gulin Bubian*. Zhonghua Book Company, Beijing, China, 2017. ISBN 9787101124132. In Chinese, 2 volumes.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Zhikai Hu, {Yiu Ming} Cheung, Yonggang Zhang, Zhang Peiying, and {Tang Pui} Ling. Component-level segmentation for oracle bone inscription decipherment. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Special Track on AI Alignment*, number 27 in Proceedings of the AAAI Conference on Artificial Intelligence, pp. 28116–28124. Association for the Advancement of Artificial Intelligence, April 2025. doi: 10.1609/aaai.v39i27.35030. Publisher Copyright: Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.; 39th Annual AAAI Conference on Artificial Intelligence, AAAI 2025 ; Conference date: 25-02-2025 Through 04-03-2025.

Caoshuo Li, Zengmao Ding, Xiaobin Hu, Bang Li, Donghao Luo, Xu Peng, Taisong Jin, Yongge Liu, Shengwei Han, Jing Yang, Xiaoping He, Feng Gao, AndyPian Wu, SevenShu, Chaoyang Wang, and Chengjie Wang. Oracleagent: A multimodal reasoning agent for oracle bone script research, 2025a. URL https://arxiv.org/abs/2510.26114.

Caoshuo Li, Zengmao Ding, Xiaobin Hu, Bang Li, Donghao Luo, AndyPian Wu, Chaoyang Wang, Chengjie Wang, Taisong Jin, Seven Shu, Yunsheng Wu, Yongge Liu, and Rongrong Ji. Oraclefusion: Assisting the decipherment of oracle bone script with structurally constrained semantic typography. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19893–19902, October 2025b.

Jing Li, Qiu-Feng Wang, Kaizhu Huang, Xi Yang, Rui Zhang, and John Y. Goulermas. Towards better long-tailed oracle character recognition with adversarial data augmentation. *Pattern Recognition*, 140:109534, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109534. URL https://www.sciencedirect.com/science/article/pii/S0031320323002340.

Shiyu Liu. Evidence-grounded oracle bone script understanding: Obb benchmark and obs-prtf framework. In De-Shuang Huang, Chuanlei Zhang, Wei Chen, Bo Li, Qinhu Zhang, Wenzheng Bao, Yijie Pan, and Prashan Premaratne (eds.), *Advanced Intelligent Computing Technology and Applications*, pp. 233–245, Singapore, 2027. Springer Nature Singapore. ISBN 978-981-92-3510-0.

Yuliang Liu, Haisu Guan, PengJie Wang, Xinyu Wang, Jinpeng Wan, Kaile Zhang, Handong Zheng, Xingchen Liu, Zhebin Kuang, Huanxin Yang, Bang Li, Yongge Liu, Lianwen Jin, and Xiang Bai. Alphaoracle: Oracle bone script decipherment via human-workflow-inspired deep learning. *The Innovation*, pp. 101462, June 2026. ISSN 2666-6758. doi: 10.1016/j.xinn.2026.101462.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Jiapeng Wang, YiFan Zhang, Zhuoma GongQue, Chong Sun, Yida Xu, Yadong Xue, Ye Tian, Zhimin Bao, Lan Yang, Chen Li, and Honggang Zhang. V-oracle: Making progressive reasoning in deciphering oracle bones for you and me. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20124–20150, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.986. URL https://aclanthology.org/2025.acl-long.986/.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Haizhao Sun, Shiyu Li, Kun Ji, Ming Wu, and Chuang Zhang. Evoacd: A balanced benchmark with relation-aware graph learning for evolution-aware archaic chinese character deciphering. In *Chinese National Conference on Computational Linguistics (CCL)*, 2026. URL https://cips-cl.org/static/CCL2026/en/cclProgram/accepted/index.html. Accepted poster paper.

Zheng Wang, Haoran Chen, Haoxuan Qin, Zhipeng Wei, Tianwen Qian, and Cong Bai. Think, then verify: A hypothesis-verification multi-agent framework for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33784–33793, June 2026.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp.

11809–11822. Curran Associates, Inc., 2023a. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Xiaosui Yao. *Yinxu Jiagu Keci Leizuan*. Zhonghua Book Company, Beijing, China, 1989. In Chinese.

Xingwu Yu. *Jiaguwen Zi Gulin*. Zhonghua Book Company, Beijing, China, 1996. ISBN 9787101014303. In Chinese, 4 volumes.

Jianing Zhang, Runan Li, Honglin Pang, Ding Xia, Zhou Zhu, Qian Zhang, Chuntao Li, and Xi Yang. Specializing large models for oracle bone script interpretation via component-grounded multimodal knowledge augmentation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35221–35236, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1626. URL https://aclanthology.org/2026.acl-long.1626/.

A Dataset Construction and Evaluation Protocol

A.1 Held-Out Character Pool

We construct the evaluation pool by separating the class IDs used to train the proposal experts from the oracle-bone-script (OBS) categories available in EVOBC (Guan et al., 2024). EVOBC contains 13,714 class IDs across all historical periods, among which 1,762 contain at least one OBC image. After removing the 1,028 class IDs used to train the proposal experts, 734 held-out candidate classes remain, containing 18,614 candidate OBC images. These class IDs are disjoint from the proposal-expert training IDs.

The subsequent data-processing status should not be interpreted as a single automatic filtering pipeline. We initially prepared a working queue of 150 candidate classes, including five development classes and 145 ordinary candidates. At the time of dataset construction, 66 classes had completed the current image–text processing workflow and entered the release ledger, including 59 classes from the original queue and seven later-added reserve candidates. Among these processed classes, 44 satisfy the current evaluation criteria and are included in the test set. The remaining unprocessed candidate classes were not rejected on quality grounds and therefore should not be interpreted as filtered-out examples.

A.2 Evaluation-Case Construction

A candidate class is included in the current evaluation set only when it can be aligned with a corresponding entry in *Yinxu Jiagu Keci Leizuan (Yao, 1989)*, the target glyph crop, local inscriptional context, and reference reading can be reliably aligned, and at least five distinct usable source groups are available. Two source groups are reserved for evaluation, while at least three groups remain available for evidence retrieval. Duplicate records, records that cannot be aligned reliably, and records that cannot be safely separated between evaluation and retrieval are excluded.

Each of the 44 evaluation classes contributes exactly two test instances, producing 88 cases in total. The two instances of the same class originate from different source groups. Importantly, the target glyph images used for evaluation are crops of transcribed or printed glyph forms from *Yinxu Jiagu Keci Leizuan*; they are not the original EVOBC target images and are not produced by OCR over raw rubbing images.

Each runtime test instance contains a target glyph image g, an inscriptional context c in which the target position is replaced by `[UNK]`, and an internal source-group identifier used only for leakage control. Reference readings are stored privately and are never exposed to the controller, proposal experts, or evidence tools during inference.

A.3 Reference Readings and Reading-Unseen Subset

The evaluation target is the acceptable character-reading or character-identity string rather than a semantic-role annotation. For instance n, we denote the accepted reference-reading set as

$$Y_n^\star = \{y_{n,1}^\star, \ldots, y_{n,m_n}^\star\}.$$

Among the 88 evaluation instances, 62 contain one accepted reading and 26 contain two accepted readings. Multiple accepted strings may correspond to historical forms, variants, or their established modern correspondences. The current dataset does not provide reference labels for place-name status, verbal usage, clan names, part of speech, or other contextual semantic roles; consequently, these properties are used as reasoning evidence but are not scored as independent prediction targets.

Although proposal-expert training IDs and evaluation IDs are disjoint, 20 evaluation instances from 10 classes share at least one accepted reading string with a training-class name. We therefore distinguish the full class-ID-disjoint evaluation set from a stricter reading-unseen subset containing 68 instances for which

$$Y_n^\star \cap \mathcal{Y}_{\text{train}} = \varnothing,$$

where $\mathcal{Y}_{\text{train}}$ denotes the set of character-name strings associated with proposal-expert training classes.

Table 4: Dataset construction and evaluation statistics. The EVOBC pool defines held-out class IDs, whereas the final evaluation glyph crops are independently constructed from corresponding entries in *Yinxu Jiagu Keci Leizuan*.

Item	Count
EVOBC full-period classes	13,714
Classes containing at least one OBC image	1,762
Proposal-expert training class IDs	1,028
Held-out candidate classes	734
Held-out candidate OBC images	18,614
Preprocessed working-queue classes	150
Classes entering the current release ledger	66
Final evaluation classes	44
Evaluation instances	88
Instances per class	2
Visible contextual records	699
Instances with one accepted reading	62
Instances with two accepted readings	26
Strict reading-unseen instances	68

A.4 Context-Corpus Leakage Control

The contextual evidence corpus contains 699 non-duplicate records derived from *Yinxu Jiagu Keci Leizuan*. Evaluation source groups are globally removed from this visible corpus before inference. Consequently, a source group used to construct a test instance cannot simultaneously appear as contextual evidence for any test case. The retrieval corpus is additionally deduplicated so that repeated copies of the same inscriptional record do not contribute multiple independent evidence items.

B Additional Qualitative Examples

Figure 3 illustrates a representative OracleResolve trajectory. The initial morphology-based hypothesis set favors [A], while [B] and another candidate remain plausible alternatives. Contextual evidence subsequently weakens the initial preference and promotes an alternative candidate, but the resulting closed-set hypotheses remain insufficiently supported. OracleResolve therefore reopens the hypothesis space through open-candidate retrieval, introducing a new candidate [D]. Newly retrieved philological evidence then provides stronger support for [D], which becomes the final top-ranked interpretation. This example illustrates the two recovery mechanisms targeted by OracleResolve: revising an incorrect preference within the existing hypothesis set and expanding the hypothesis space when the current candidates remain inadequate.

C Proposal Experts and Initial Hypothesis Construction

C.1 Glyph Expert

The Glyph Expert provides global-shape and visual-similarity cues. A ResNet-101 encoder (He et al., 2016) extracts the 2,048-dimensional average-pooled visual representation of the target glyph. The representation is passed through a prototype head with dimensions $2048 \to 256 \to 256 \to 256$, and the final embedding is ℓ_2 normalized. Candidate neighbors are retrieved from the shared proposal-expert training gallery using cosine similarity. For each training class ID, only its nearest image is retained, preventing a class with many images from dominating the neighbor set. The top 20 distinct training classes are retrieved, and the top five are exposed to the candidate-generation model during initial hypothesis construction.

The retrieved visual neighbors are not treated as final predictions. The candidate-generation model receives the target glyph, the top-five neighbor glyphs, their associated training readings, cosine

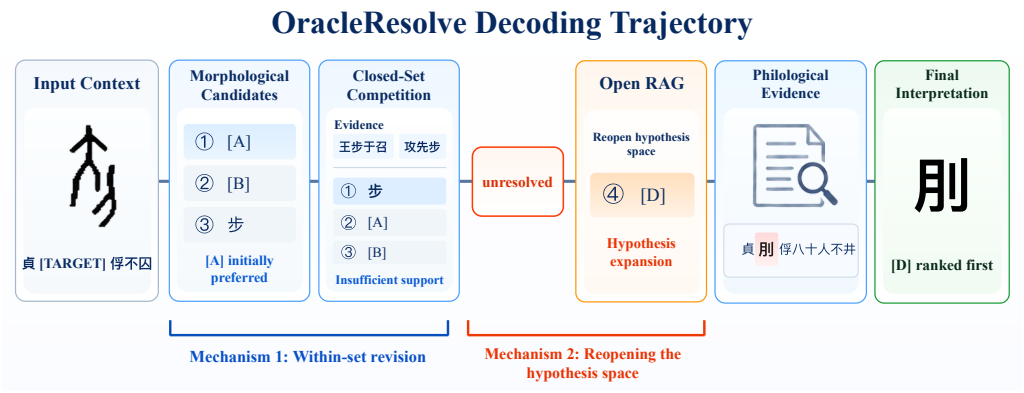


Figure 3: A complete recovery path from preference correction to open-ended hypothesis expansion.

Table 5: Configuration of the two proposal experts used to construct the initial hypothesis set.

Expert	Backbone	Primary output	Candidate cue
Glyph Expert	ResNet-101	256-D normalized embedding	Global shape / similarity
Component Expert	ResNet-18	796-label sigmoid vector	Component composition

similarities, and the public character inventory, and generates three to five distinct single-character candidate interpretations. The local inscriptional context c is not used during this initialization stage.

C.2 Component Expert

The Component Expert provides complementary compositional cues. It uses a ResNet-18 backbone (He et al., 2016) with 224×224 RGB inputs normalized using ImageNet statistics. The network predicts 796 component labels using independent sigmoid outputs. For each target glyph, we retain the ten highest-scoring components and additionally record all component predictions with scores greater than or equal to 0.5. The current model does not perform component segmentation or predict spatial relationships among components.

The candidate-generation model receives the target glyph, predicted components and their scores, and the public character inventory, and independently proposes three to five character-level interpretations. As with the Glyph Expert, the local inscriptional context is not provided during initialization.

C.3 Candidate Fusion and Initial Scores

The two candidate branches are assigned equal total weight. For a candidate c appearing at rank r in branch b, the branch contribution is

$$w_b(c) = \frac{1}{2r}.$$

If the same character is proposed by both branches, its contributions are summed,

$$w(c) = \sum_b w_b(c).$$

Candidates are ranked by $w(c)$; Unicode order is used only as a deterministic tie-breaker. Duplicate candidate strings are merged, and at most five candidates are retained. Their weights are normalized into an initial prior

$$p(c_i) = \frac{w(c_i)}{\sum_j w(c_j)},$$

and OracleResolve initializes the hypothesis score as

$$s_i^0 = \log p(c_i).$$

```
## Case one:

### Input

![Target glyph](target.png)

Input context:

> …ト□龜一ト五…

The square marks the unknown target. Ellipses preserve omissions in the original input. Inscription excerpts are retained in Chinese to avoid adding an unverified translation.

The initial output order was **右, 疋, 羽, 言, 有**. 右 was listed first, initially tied in score with 疋. 習 was absent from the initial candidate set.

### Key steps

| Stage | Evidence and model judgment | Candidate change |
|---|---|---|
| First glyph comparison | The model compared a 右-like structure with a 疋-like outline. A comparison glyph labeled 帚 was judged to bear a rough resemblance to the target's branching strokes and box-shaped lower part. | 帚 was added in third place; 右 remained first. |
| Second glyph comparison | The model sought further morphological features that could distinguish 右 from 疋. The new materials did not change the candidate order. | 右, 疋, and 帚 remained the first three candidates. |
| Switch to contextual retrieval | A search for the inscription pattern ト龜一ト五 returned parallel examples with 習 in the relevant position. | 習 was introduced from outside the initial set and rose to first place after the evidence was reassessed. |

Key inscription examples:

> …ト習龜一ト五…
>
> 習一ト五于…

[View the first inscription image]
[View the second inscription image]

The expansion module noted that the first example places 習 between ト and 龜. It also treated the target's branching upper strokes and box-shaped lower part as possible clues for further comparison. The evidence judge assigned support to 習 and negative contributions to the older candidates. These are model judgments: another inscription's use of 習 does not itself prove that the alternative readings must be wrong in the target instance.

### Final outcome

The final output order was:

**習 → 右 → 疋 → 帚 → 羽 → 言 → 有**

習 became the top-ranked candidate and matched the draft reference reading. The run remained unresolved because the evidence did not meet the confirmation criteria.

This case illustrates how contextual retrieval can supply a reading missing from the initial candidate set when glyph comparison has not resolved the problem. It is **recovery through candidate expansion**, rather than reranking a correct answer already present in the initial set.

### Suggested figure caption

> The initial list placed 右 first while retaining alternatives including 疋 and 羽. Two rounds of glyph comparison introduced 帚 but did not change the leading candidate. Contextual retrieval then returned parallel examples, including "…ト習龜一ト五…" and "習一ト五于…", prompting the system to introduce 習 from outside the initial candidate set. Evidence updates moved 習 to first place. The trace illustrates how contextual material can expand an inadequate candidate set and change the preferred reading; the final interpretation remains subject to verification.
```

Figure 4: Additional representative cases.

These scores are evidence-aggregation scores rather than calibrated correctness probabilities. All matched reasoning baselines receive exactly the same initial hypothesis set $\mathcal{H}_0$ and initial evidence state $\mathcal{E}_0$. The Direct-LLM baseline does not access these proposal-expert outputs.

D Evidence Resources and Retrieval

OracleResolve dynamically queries three external evidence tools after initialization: contextual evidence, diachronic evidence, and philological evidence. Morphological information is supplied by the two proposal experts during initialization and is therefore not exposed as a separate dynamically selected evidence tool.

D.1 Context Tool

The Context Tool operates over the 699-record held-out context corpus described in Appendix A. The central planner generates a retrieval query, which is tokenized into Chinese characters and alphanumeric tokens. Records are ranked by token-overlap ratio with an additional exact-substring

```
## Case two:

### Input

![Target glyph](target.jpg)

Input context:

> 其遭大□

The initial output order was **伐, 疾, 失, 戈, 戍**. 伐 was listed first, initially tied in score with 疾. 風 was absent from the initial
candidate set.

### Key steps

| Stage | Evidence and model judgment | Candidate change |
|---|---|---|
| First-round contextual retrieval | The model initially sought to distinguish a warfare interpretation from an illness interpretation, but
retrieved parallel examples reading 其遭大風. | 風 was added and immediately rose to first place after the evidence update. |
| Second-round glyph check | With a new leading candidate, the model shifted to comparing the predicted glyph features of 風 and 伐 and actively
sought counterevidence. | The retrieved comparison examples yielded no judgments sufficient to change the ranking; 風 remained first. |
| Third-round glyph check | The model examined further similar glyphs. | No additional effective discriminating evidence changed the ranking. |

Key inscription example:

> 其遭大風

Retrieval returned two records with this same transcription: [first inscription image]· [second inscription image]. The run scored them as
different provisional source groups. Their scholarly source independence has not been verified; they cannot be presented as two independently
established corroborations.

The model judged 風 compatible with the target position after 大 and introduced it as a candidate. The subsequent glyph checks did not overturn
this preference, but neither provided independent visual confirmation. The image is small, and the model explicitly retained the limitation that
the glyph alone could not establish the reading.

### Final outcome

The final output order was:

**風 → 伐 → 疾 → 失 → 戈 → 戍**

風 became the top-ranked candidate and matched the draft reference reading. The run nevertheless remained unresolved.

This case illustrates how contextual material can move the system beyond its original competition between 伐 and 疾, redirecting later
investigation toward a comparison of the new candidate with an earlier one. Failure to find evidence overturning the new candidate is not proof
that it is correct.

### Suggested figure caption

> The initial candidates were led by 伐 and 疾 and did not include 風. Contextual retrieval in the first round returned parallel examples reading
"其遭大風", prompting the system to introduce 風 and rank it first. Two subsequent rounds examined glyph differences between 風 and 伐 but
produced no judgments sufficient to change the ranking. 風 therefore remained first, while the glyph correspondence and source independence
awaited verification. The trace illustrates how the introduction of a candidate through new evidence can redirect subsequent investigation.
```

Figure 5: Additional representative cases.

bonus. We do not use embedding retrieval, BM25, or an LLM reranker in the current implementation. Each tool invocation returns at most two evidence records.

D.2 Diachronic Tool

The Diachronic Tool is instantiated using EvoGraph trained on EvoACD (Sun et al., 2026). Given a candidate interpretation, it returns cross-period character-form evidence used to examine whether the proposed reading is compatible with known diachronic correspondences. The current paper treats EvoGraph as a fixed external evidence module rather than retraining it as part of OracleResolve.

D.3 Philological Tool

The Philological Tool queries a curated evidence corpus derived from *Jiaguwen Zigu Lin (Yu, 1996)* and *Jiaguwen Zigu Lin Bubian (He, 2017)*. Retrieved records provide dictionary entries and documented scholarly discussions relevant to candidate interpretations. Because these materials are also used as runtime evidence, opinions appearing in the same corpus are not treated as independent expert evaluation labels in our experiments.

Table 6: Evidence resources available after initial hypothesis construction.

Tool	Evidence source	Retrieval mechanism	Output
Context	*Yinxu Jiagu Keci Leizuan*	Token overlap + substring bonus	Inscriptional examples
Diachrony	EvoGraph / EvoACD	EvoGraph inference	Historical forms
Philology	*Jiaguwen Zigu Lin* / *Jiaguwen Zigu Lin Bubian*	Curated lookup	Scholarly records

Table 7: Controller and reasoning-budget configuration used in the main experiments.

Setting	Value
Controller model ID	`gpt-6-sol`
Reasoning effort	`high`
Maximum output tokens	8192
Temperature	Not explicitly set
Random seed	Not set
Image detail	`high`
Maximum reasoning rounds	3
Maximum evidence-tool calls	3
Evidence items per tool call	2
Maximum logical-model calls	8
Maximum execution attempts per stage	2
Stability rounds	2
Top-1 relative-support threshold	0.75
Top-1 margin threshold	0.20

E Controller, Reasoning Budget, and Decision Rules

E.1 Controller Configuration

The candidate-generation model, central planner, Evidence Judge, and Generic-Agent baseline use the same third-party Responses-compatible API exposing the model identifier `gpt-6-sol`. The service is queried with reasoning effort `high`, a maximum output length of 8,192 tokens, and high-detail image inputs. Temperature and random seed are not explicitly set, and requests use `store=false`. The API provider does not expose a date-stamped model snapshot. We therefore report the returned model identifier and request configuration but cannot further resolve the backend model version.

The planner, Evidence Judge, and hypothesis-revision functions are functional roles executed by the same central LLM controller rather than separate autonomous agents. Proposal experts and evidence modules are external callable components.

E.2 Reasoning Budget

Each evaluation instance is limited to at most three reasoning rounds and three evidence-tool calls. Each evidence-tool call returns at most two records. The controller may make at most eight logical-model calls in total, and each stage permits at most two execution attempts. Initial hypothesis generation is performed before the controller loop and is excluded from this budget.

E.3 Termination and Expansion

A normal evidence-sufficient termination requires the complete candidate ranking to remain unchanged for two consecutive rounds, the Top-1 relative support to exceed the fixed termination threshold, the Top-1 margin over the second-ranked candidate to exceed the fixed margin threshold, the Top-1 candidate to receive substantive support from at least two independent source groups,

no unresolved substantive counterevidence against the Top-1 candidate, and at least one completed challenge step. The exact threshold defining substantive evidence is fixed on the development set before test evaluation.

If every active candidate receives substantive counterevidence from at least two independent source groups and none receives substantive support, the state is marked ALL-REJECTED and hypothesis expansion is triggered. Additional low-confidence expansion conditions, if used in the final experiments, are fixed on the development classes and reported in Table 7. If the reasoning budget is exhausted before a stopping condition is met, OracleResolve preserves the final ranking but marks the result as unresolved rather than forcing an evidence-sufficient conclusion.

F Open-Candidate Hypothesis Expansion

When the current hypothesis set is judged insufficient, OracleResolve invokes the open-candidate generator G_{exp}. The generator receives the target glyph, the current hypothesis state, accumulated evidence, and the available contextual information, and proposes candidate readings that are not already present in the active hypothesis set.

In the reported experiments, each expansion call proposes at most two new single-character candidates. Each proposed candidate must be grounded in a non-initial evidence item and accompanied by a verifiable source quotation. Retrieved expansion evidence is passed through the same Evidence Judge used during ordinary reasoning and is not automatically treated as support for the newly proposed candidate.

After duplicate removal, the new candidates and their associated evidence are merged into the persistent hypothesis state. The expanded hypotheses then participate in subsequent evidence acquisition and revision under the same reasoning protocol as the original candidates.

G Baseline and Ablation Definitions

G.1 Main Baselines

Direct-LLM receives the target glyph, local context, and public character inventory and directly predicts candidate readings without accessing the local proposal experts or evidence tools.

Fixed-Pipeline receives the same initial hypothesis set and evidence state as OracleResolve, and queries the same external evidence sources in a fixed order: Context, Diachrony, and Philology. It uses the same Evidence Judge but does not perform disagreement-driven tool selection.

Generic-Agent receives the same initial hypotheses, evidence resources, and reasoning budget as OracleResolve and may freely select among the available evidence tools. It does not use OracleResolve's explicit competing-hypothesis evidence matrix and score-based hypothesis-revision mechanism.

OracleResolve uses the full proposed framework, including persistent competing hypotheses, disagreement-driven evidence acquisition, supporting and contradicting evidence memory, explicit score revision, replanning, and open-candidate hypothesis expansion.

G.2 Ablation Variants

The Top1-Only variant discards competing alternatives and reasons only over the currently highest-ranked candidate. The No-Contradiction-Memory variant removes persistent counterevidence memory while retaining supporting evidence. The No-Replan variant prevents newly revised hypothesis states from changing the previously selected investigation plan. The Fixed-Order variant replaces disagreement-driven evidence selection with the same predefined evidence sequence used by Fixed-Pipeline.

Table 8: Summary of main baselines and OracleResolve ablations.

Method	Shared $\mathcal{H}_0$	Dynamic tools	Competing hypotheses	Expansion
Direct-LLM	No	No	No	No
Fixed-Pipeline	Yes	No	Yes	Yes
Generic-Agent	Yes	Yes	Implicit	Yes
OracleResolve	Yes	Yes	Yes	Yes
Top1-Only	Yes	Yes	No	Yes
No-Contradiction-Memory	Yes	Yes	Yes	Yes
No-Replan	Yes	Yes	Yes	Yes
Fixed-Order	Yes	No	Yes	Yes

H Evaluation Metrics

For each test instance n, let $\widehat{Y}_n^{(k)}$ denote the first k ranked predictions and $Y_n^\star$ the accepted reference-reading set. Top-k accuracy is defined by

$$\mathrm{Acc@}k = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left[\widehat{Y}_n^{(k)} \cap Y_n^\star \neq \varnothing\right].$$

For mean reciprocal rank, we use the highest-ranked accepted reading:

$$\mathrm{MRR} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\min_{y\in Y_n^\star}\mathrm{rank}_n(y)}.$$

For cases where at least one accepted reading is contained in $\mathcal{H}_0$ but the initial Top-1 prediction is incorrect, Rank Recovery is

$$\mathrm{RR}_{\mathrm{rank}} = \frac{\#\{\text{incorrect initial Top-1 corrected at termination}\}}{\#\{\text{incorrect initial Top-1 with } Y^\star \cap \mathcal{H}_0 \neq \varnothing\}}.$$

Erroneous Revision measures the opposite failure mode:

$$\mathrm{ER} = \frac{\#\{\text{initially correct Top-1 changed to an incorrect final Top-1}\}}{\#\{\text{initially correct Top-1}\}}.$$

For cases where $Y^\star \cap \mathcal{H}_0 = \varnothing$, Expansion Recall measures whether an accepted reading is introduced at any point by hypothesis expansion:

$$\mathrm{RR}_{\mathrm{exp}} = \frac{\#\{\text{accepted reading absent initially but introduced after expansion}\}}{\#\{Y^\star \cap \mathcal{H}_0 = \varnothing\}}.$$

We additionally report Expansion Top-1, the fraction of these cases for which an expansion-introduced accepted reading becomes the final Top-1 prediction. Evidence efficiency is reported as the average number of evidence-tool calls per test instance.

I Core Prompt Templates

For readability, we provide compact English renderings of the core instruction templates underlying OracleResolve. The reported experiments were conducted using the original Chinese instructions; therefore, the English text below is intended to document the instructional logic rather than to serve as a behaviorally validated translation. Case-specific payload fields, JSON schemas, tool-interface metadata, and the exact original-language prompts are provided in the supplementary material.

Common system instruction. All controller calls share a common system instruction that requires evidence-grounded and auditable reasoning:

```
You are an oracle-bone evidence research assistant. Treat only the
↪ supplied original materials as evidence; candidate interpretations
↪ and model predictions are not facts.

Do not fabricate sources, inscription examples, components, historical
↪ periods, or experimental observations. Distinguish character
↪ identity, variant relationships, and usage in the current sentence.
↪ Relevance does not imply support, and failure to retrieve relevant
↪ material does not constitute counterevidence.

The same material may support multiple candidates. Explicitly indicate
↪ when a judgment cannot be made. Return only the requested structured
↪ output; explanations should be brief and auditable.
```

Initial candidate proposal. The Glyph and Component proposal branches share the same open-ended candidate-generation objective. They differ only in the auxiliary perceptual cues supplied to the model.

```
Assess possible readings of the oracle-bone glyph and propose 3--5
↪ distinct single-character candidates, ordered by plausibility. This
↪ is not closed-set classification, and candidates should not be
↪ restricted to the supplied reference classes.

Give only brief, auditable justifications and do not fabricate
↪ archaeological or philological evidence. The accompanying public
↪ character vocabulary is only a shared reference list and does not
↪ identify the answer; characters outside the list remain allowed.

Return a structured list of candidate readings together with short
↪ rationales and limitations.
```

For the Glyph branch, retrieved training-set neighbors are explicitly described as visual references rather than known identities of the target, and similarity scores are not interpreted as reading probabilities. For the Component branch, predicted components are explicitly described as model outputs rather than verified segmentations or spatial relations.

Disagreement-aware planning. The planner is instructed to select evidence according to an unresolved distinction among the current hypotheses rather than simply accumulating additional support for the leading candidate:

```
Choose one useful next evidence query. Identify a specific unresolved
↪ disagreement between the current candidates and formulate an
↪ observation that could distinguish them.

For the competing hypotheses, state different testable predictions
↪ whenever applicable. Shared features are not discriminating evidence.
↪ Actively search for counterevidence when necessary.

Return only one action at the current step; subsequent actions will be
↪ replanned after new observations. Previous assessments are model
↪ judgments rather than new facts, and unresolved questions should not
↪ be treated as resolved merely because the ranking is stable.

Do not fabricate observations or sources.
```

Open-candidate hypothesis expansion. When the current hypothesis space is judged insufficient, the expansion instruction explicitly permits the introduction of readings outside the active candidate set:

```
Check whether the current candidate set omits a reading that better
↪  explains the target glyph and the available context. Based only on
↪  the actual materials already obtained, propose at most two distinct
↪  new single-character readings; returning no new candidate is allowed.

Do not identify the target as a character merely because that character
↪  appears in a retrieved source. Explain how the target evidence
↪  relates to each proposed reading.

Each new candidate must be grounded in a non-initial evidence item and
↪  accompanied by a verifiable source quotation. Do not propose
↪  characters already contained in the current hypothesis set, and do
↪  not fabricate sources.
```

Evidence judgment. The Evidence Judge evaluates retrieved evidence against each active hypothesis independently:

```
Judge each original evidence item against each candidate independently.
↪  For every evidence--candidate pair, output a stance from {support,
↪  contradict, neutral, unjudgeable}, together with strength, relevance,
↪  a source quotation when applicable, and a brief reason.

Support or contradiction requires a verbatim quotation that can be
↪  located in the corresponding evidence item. Relevance alone does not
↪  imply support, and the same evidence may support more than one
↪  candidate.

Retrieved examples associated with a candidate are analogical evidence
↪  and do not by themselves establish that the input target has that
↪  identity. Use unjudgeable when the available materials are
↪  insufficient or the historical usage is unclear. Do not omit
↪  evidence--candidate combinations.
```

The full original-language prompts additionally contain structured input fields, output schemas, tool descriptions, length constraints, and condition-specific controller modifications. These implementation details and the exact prompts used in the reported experiments are included in the supplementary material.