LEARNING FROM INCOMPLETE AND UNALIGNED LINEAGES FOR ORACLE BONE SCRIPT DECIPHERMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Oracle Bone Script (OBS) decipherment seeks the established modern counterpart of an ancient glyph. Related forms preserved across later script periods provide chronologically ordered historical evidence that can help bridge this large visual gap. Yet a cascade of models trained between consecutive stages performs worse than a single model that maps OBS directly to the modern character. We found that the difficulty lies not in the existence of intermediate evidence but in how it is used. For most lineages are incomplete, and the variants of a character at one stage are not paired with those at the next, so each step of the cascade is trained on fragments and restricts cross-period context. We introduce ChronoOBS, which jointly models the six script periods as a chronologically ordered sequence with a shared autoregressive transformer and diffusion head. Missing stages are represented by placeholders, Dynamic Optimal Transport Matching (DOTM) constructs soft cross-period correspondences between unpaired variants, and multi-path decoding aggregates predictions over different subsets of intermediate periods. Under the same underlying architecture, the full ChronoOBS pipeline reaches 24.4% Top-1 accuracy on held-out character categories, compared with 21.8% for direct training under the same architecture and 20.5% for the strongest directly comparable prior image-generation method. On known character categories with period-specific references, the generated intermediate glyphs preserve character identity substantially better than cascaded trajectories. Code and models will be made available.

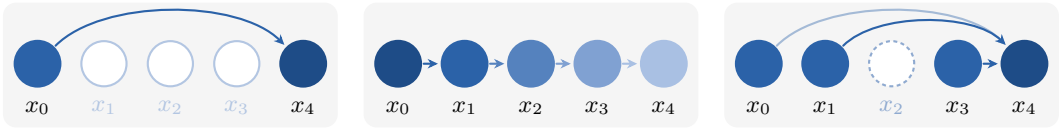
1 INTRODUCTION

Recent progress in AI has shown that large gaps are often easier to bridge through intermediate steps, from chain-of-thought reasoning in language models (Wei et al., 2022; Guo et al., 2025) to iterative denoising in diffusion models (Ho et al., 2020), typically when those steps are supplied or constructed, or when they form a coherent and well-aligned sequence. However, many problems in the sciences and humanities offer neither convenience: their intermediate steps survive only as fragmentary records of a past process, with many of the connections among them missing or unclear. The decipherment of Oracle Bone Script (OBS), the earliest mature form of Chinese writing, exemplifies this challenge. Roughly two-thirds of the more than 4,500 known OBS characters remain undeciphered despite over a century of study (Guo & Hu, 1978), and the historical forms spanning the three millennia from OBS to modern Chinese survive only partially, while the relationships among variants of the same character across different script periods are often ambiguous. This raises a broader question: *how can a model bridge a large gap when the intermediate observations are incomplete and unaligned?*

In OBS, the steps in between are attested historical scripts linking each oracle-bone glyph to its modern form. Computational OBS decipherment aims to predict this modern counterpart, represented in Regular Script (RS) (Zhang et al., 2021; Guan et al., 2024b). Paleographic analysis often compares related character forms across historical script traditions and periods, using structural continuity and change as part of the evidence for decipherment (Qiu, 2000), but for most characters some stages are missing from the record. Learning from such partially preserved lineages poses two challenges that extend beyond OBS and that standard step-by-step learning does not jointly address.

- **Incomplete intermediate steps.** Supervising intermediate steps requires them to be recorded for each example, as reasoning traces or paired data between adjacent stages (Wei et al., 2022). In

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.



(a) Direct (b) Cascade (c) Joint (ours)

Figure 1: Comparison of learning paradigms. (a) Direct mapping ignores the intermediate stages. (b) Cascade is restricted to adjacent-period evidence, with missing observations breaking supervision links. (c) Joint modeling learns the entire lineage as one trajectory, allowing every available intermediate observation to inform prediction even when other steps are missing (dashed).

OBS, most characters are unattested at one or more intermediate periods, so such chains are broken for most training examples.

- **Unaligned observations.** Even where intermediate states are observed, they often arrive as unordered collections rather than linked sequences, as when developmental trajectories must be inferred from unpaired population snapshots (Schiebinger et al., 2019). In OBS, a character may have several variant forms in each period, yet no record states which variant developed into which.

When intermediate observations are incomplete and unaligned, common strategies either ignore them and learn the end-to-end mapping directly (Fig. 1(a)), or use them only as auxiliary training targets, or keep only the fragments that are complete and cascade models trained between adjacent stages (Fig. 1(b)). Each of these strategies leaves part of the available evidence unused. Auxiliary targets shape the representation but leave each prediction unconditioned on the lineage. In fact, existing OBS decipherment methods also follow either a direct or a cascaded route. Direct methods map an oracle-bone glyph to its modern counterpart through component analysis, multimodal reasoning, or image generation (Wang et al., 2024a; Peng et al., 2025; Guan et al., 2024b), leaving the intermediate periods unused. Evolution-based methods cascade separately trained stage-to-stage models between adjacent periods (Chang et al., 2022), but each model learns only from characters observed at both of its periods, and sees no evidence beyond its own pair. Missing periods further interrupt adjacent-pair supervision, leaving available forms on either side disconnected during training. Under the same architecture, we find that such a cascade performs even worse than direct mapping, while auxiliary targets change accuracy by less than one point.

These observations motivate our central view: the entire lineage should be learned jointly (Fig. 1(c)), linking all available observations into latent evolutionary trajectories. Evidence from each recorded period can then inform subsequent evolutionary steps, even when some periods are missing.

To this end, we propose ChronoOBS, which learns all script periods jointly as one chronologically ordered sequence with a shared autoregressive-diffusion model. Missing periods remain in position as placeholders, so incomplete lineages can contribute supervision without forcing a complete chain. For unaligned variants, we propose Dynamic Optimal Transport Matching to assign soft probabilities to alternative cross-period correspondences and sample coherent training trajectories, while multi-path decoding aggregates predictions across period subsets. On held-out deciphered categories, the full pipeline reaches 24.4% Top-1 accuracy, compared with 21.8% for direct training under the same architecture. Independent verification on known categories shows substantially stronger preservation of character identity than the cascade. The broader formulation concerns partially observed and unpaired intermediate states, but the empirical validation in this work is limited to OBS. Our contributions are threefold:

- We frame OBS decipherment as learning from partially preserved lineages, where intermediate evidence is incomplete, unaligned, and without verifiable targets, and show that cascading through such evidence performs worse than ignoring it, whereas modeling it jointly helps.
- We propose ChronoOBS, a joint autoregressive-diffusion model over script periods, with Dynamic Optimal Transport Matching (DOTM) aligning unpaired variants along entire trajectories and multi-path decoding aggregating predictions over subsets of periods.
- Through controlled comparisons, we separate the effects of training paradigm, intermediate conditioning, stochastic ensembling, and path diversity. ChronoOBS remains stronger under matched final-output budgets, degrades gracefully as intermediate records are removed, and produces substantially more identity-consistent trajectories than cascading on known categories.

2 Related Work

2.1 Oracle Bone Script Decipherment

Machine learning has been applied to the decipherment and restoration of ancient scripts (Sommerschield et al., 2023), including lost languages such as Ugaritic and Linear B (Luo et al., 2019). Within this line, OBS decipherment identifies the modern Chinese character that corresponds to an oracle-bone glyph. Early work on OBS focused on closed-set recognition (Huang et al., 2019; Han et al., 2020; Zhao et al., 2022; Wang & Deng, 2024), which does not cover the unseen categories that decipherment targets. Direct decipherment methods predict the modern counterpart from the oracle-bone glyph alone, through retrieval, component analysis, multimodal reasoning, or glyph generation (Zhang et al., 2021; Wang et al., 2024a; Lin et al., 2026; Qiao et al., 2025; Wang et al., 2026; Peng et al., 2025; Guan et al., 2024b; Wu et al., 2025; Li et al., 2025). Evolution-based methods use the intermediate scripts. Wang et al. (2022) compare structural similarity across five periods, and Sundial-GAN (Chang et al., 2022) cascades adjacent-period GANs to generate intermediate forms. Such cascades learn each transition only from paired characters in adjacent periods, confining predictions to local evidence and causing missing periods to disrupt the entire evolutionary chain. ChronoOBS addresses this limitation by jointly learning all periods in one model, so incomplete lineages still contribute to training and each prediction can condition on the available forms.

2.2 Learning from Incomplete and Unaligned Intermediate Stages

Intermediate stages are a common means of bridging large gaps, as in gradual domain adaptation (Kumar et al., 2020), chain-of-thought supervision (Wei et al., 2022), and diffusion models (Ho et al., 2020). Such stages may be collected, generated by the model, or synthesized by a fixed noising process. Ancient inscription restoration recovers missing text from surviving context (Assael et al., 2019; 2022), whereas in OBS decipherment, we ask how partially preserved intermediate forms can help identify the modern counterpart of an observed ancient glyph. When observations are also unpaired, as for cells measured only once in single-cell biology, trajectories are inferred by coupling snapshots at successive time points through optimal transport (Schiebinger et al., 2019), continuous normalizing flows (Tong et al., 2020), multi-marginal formulations (Haasler et al., 2021), or Schrödinger bridges (De Bortoli et al., 2021). Our work shares with these approaches the use of cross-stage correspondences to learn from unpaired observations. For OBS decipherment, we use Dynamic Optimal Transport Matching to assemble training trajectories from glyph variants of the same character across observed periods. These trajectories support a shared autoregressive model that learns from incomplete lineages and conditions each subsequent prediction on the available preceding forms, linking intermediate trajectory generation to modern-character prediction.

3 Method

3.1 Problem Formulation

We formalize OBS decipherment as learning from a partially preserved lineage of scripts. Consider categories observed at a chronologically ordered sequence of stages $\mathcal{P} = (p_1, \ldots, p_K)$, where categories may be missing at some stages and observations across stages are unpaired. For OBS, $\mathcal{P}$ consists of OBS, Bronze Inscriptions (BI), Warring States script (WSC), Small Seal script (SS), Clerical script (CS), and Regular Script (RS), with $K = 6$; we refer to these stages as script periods. These periods provide a chronological scaffold for historical evidence without assuming unique, linear descent between successive script traditions. Parallel and overlapping developments are abstracted at the period level; "trajectory" and "lineage" denote this model-level ordering. For character category c, the glyph variants observed in period p_k form the set $\mathcal{X}_c^k = \{X_{c,i}^k\}_{i=1}^{n_c^k}$. The indicator $\delta_c^k = \mathbf{1}[n_c^k > 0]$ specifies whether that period is available. A training trajectory is the K-position sequence $\mathbf{X}_c = (X_c^1, \ldots, X_c^K)$, where $X_c^k \in \mathcal{X}_c^k$ if $\delta_c^k = 1$ and $X_c^k = \varnothing$ otherwise. Since instance-level correspondences across periods are unobserved, $\mathbf{X}_c$ must be assembled from category-level evidence. We compare three paradigms for mapping X_c^1 (OBS) to X_c^K (RS).

Direct modeling. The direct paradigm learns a single source-to-target mapping

$$p_{\text{dir}}(X_c^K \mid X_c^1). \tag{1}$$

This bypasses the intermediate periods (Fig. 1(a)). A multi-task variant adds auxiliary losses $p(X_c^k \mid X_c^1)$ for the intermediate periods but still predicts X_c^K from X_c^1 alone.

Cascade modeling. For a complete trajectory, the cascade paradigm decomposes evolution into independently parameterized adjacent-period mappings (Fig. 1(b)),

$$p(X_c^2, \ldots, X_c^K \mid X_c^1) \approx \prod_{k=2}^{K} p_{\text{pair},k}(X_c^k \mid X_c^{k-1}). \tag{2}$$

Separate training lacks a shared representation, confines each transition to adjacent-period conditioning, and prevents models from exploiting non-adjacent evidence or incomplete lineages.

Joint evolution modeling. We jointly model all periods with a single shared model (Fig. 1(c)). For a complete trajectory, the autoregressive factorization is

$$p_\Theta(X_c^2, \ldots, X_c^K \mid X_c^1) = \prod_{k=2}^{K} p_\Theta(X_c^k \mid X_c^1, \ldots, X_c^{k-1}), \tag{3}$$

where Θ denotes the trainable parameters shared across target periods. Each prediction integrates all available observed or generated history. Incomplete trajectories retain their chronological positions and provide supervision only at available targets. This joint formulation unifies trajectory generation and final decipherment within one model.

3.2 Joint Multi-Period Evolution Framework

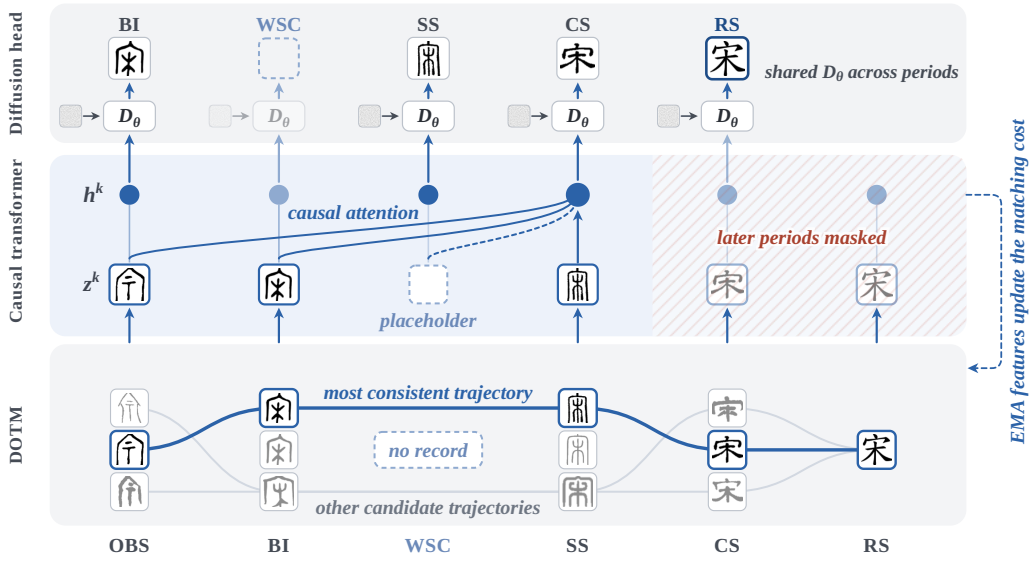


Figure 2: Overview of ChronoOBS training. DOTM samples a complete trajectory (blue) of matched variants, and a period without records (WSC) enters as a placeholder. The causal transformer attends only to earlier periods, and one diffusion head D_θ generates each next period from the preceding state. The matching cost combines a fixed path-signature distance with a distance between the model's EMA features, which is refreshed periodically.

We realize the joint factorization in Eq. 3 with an autoregressive transformer and a shared diffusion head (Fig. 2). The transformer summarizes the available history into period-specific states, the diffusion head generates the glyph of the next period from these states, and at inference, each generated glyph is re-encoded for the next prediction. Three further components address the properties of the record discussed in the introduction. Placeholder tokens and random period dropping let incomplete lineages supervise the model (see below), Dynamic Optimal Transport Matching (DOTM) aligns unpaired variants across periods (Sec. 3.3), and multi-path decoding aggregates predictions over subsets of intermediate periods (Sec. 3.4).

Evolution sequence construction. Historical records provide uneven coverage across script periods, so complete cross-period observations are rare. To handle these incomplete intermediate steps, we retain all K chronological positions and represent unobserved forms with placeholder tokens, namely learnable period embeddings. For period p_k with embedding e_k, the input token is

$$z_c^k = \begin{cases} E_{\text{img}}(X_c^k) + e_k, & \delta_c^k = 1, \\ e_k, & \delta_c^k = 0, \end{cases} \tag{4}$$

where E_{img} is the image encoder. During training, each available intermediate glyph is independently dropped from the conditioning sequence with a probability of 0.5, exposing the model to diverse incomplete histories and enabling it to operate with different subsets of periods.

Chronological autoregressive modeling. The period tokens are processed by an autoregressive transformer under a chronological attention mask,

$$(h_c^1, \ldots, h_c^K) = \text{Transformer}_\phi(z_c^1, \ldots, z_c^K), \tag{5}$$

where h_c^k summarizes the filled positions up to period p_k. The chronological attention mask (hatched in Fig. 2) permits a token at p_i to attend to p_j only if $j \le i$. This prevents later glyphs from leaking into earlier predictions while allowing each state to retain information from all preceding observed or generated glyphs.

Diffusion transformer head. Following MAR (Li et al., 2024), which pairs an autoregressive backbone with a lightweight diffusion head, we use one lightweight DiT head D_θ shared across all post-OBS periods. Conditioned on h_c^{k-1}, it generates the next glyph at p_k. Following JiT (Li & He, 2025), we predict the clean glyph but compute the loss in velocity space:

$$X_{c,t}^k = tX_c^k + (1-t)\epsilon, \quad v_{c,t}^k = X_c^k - \epsilon, \quad \widehat{X}_{c,\theta}^k = D_\theta(X_{c,t}^k, t, h_c^{k-1}), \quad \widehat{v}_{c,\theta}^k = \frac{\widehat{X}_{c,\theta}^k - X_{c,t}^k}{1-t}. \tag{6}$$

Let $\mathcal{K}_c^+ = \{k \in \{2, \ldots, K\} : \delta_c^k = 1\}$ denote the available post-OBS targets. We train with the v-loss,

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{(c,\mathbf{X}_c),\, k\in\mathcal{K}_c^+,\, t,\epsilon} \left[\| \widehat{v}_{c,\theta}^k - v_{c,t}^k \|_2^2 \right]. \tag{7}$$

Following prior work (Guan et al., 2024b), Localized Structural Sampling (LSS) applies overlapping local crops at each diffusion stage to preserve fine-grained glyph structure.

3.3 Dynamic Optimal Transport Matching

Historical records provide multiple glyph samples for each character category at each observed period, forming the variant set $\mathcal{X}_c^k = \{X_{c,i}^k\}_{i=1}^{n_c^k}$, but they rarely specify their cross-period correspondences. To recover these unaligned correspondences, we adopt optimal transport (OT), a classical approach to image-distribution matching (Rubner et al., 2000), whose transport plan assigns soft correspondences between variants. Applying standard OT independently to period pairs with fixed image costs, however, neither enforces consistency along a complete trajectory nor adapts to learned cross-period representations. We therefore propose Dynamic Optimal Transport Matching (DOTM), shown at the bottom of Fig. 2, which addresses both limitations with a static–dynamic cost and a temporally coherent joint transport plan over all observed periods.

Static–dynamic cost. To compute the matching cost between variants, we first extract static and dynamic features. Because the writing order of ancient glyphs is unknown, we use a Hilbert scan as a deterministic, locality-aware ordering heuristic and then compute normalized path-signature features, following their established use in handwriting analysis (Graham, 2013). This ordering serves as a practical representation choice, without assuming historical accuracy in stroke order. Dynamic features are normalized period-aware representations produced by an exponential-moving-average (EMA) copy of the encoder and autoregressive transformer, with each candidate placed at its period and placeholders elsewhere. Let $d_{\text{sta}}^{k,l}(i,j)$ and $d_{\text{dyn}}^{k,l}(i,j)$ denote the squared Euclidean distances between the corresponding static and dynamic features, respectively. The final cost is

$$d^{k,l}(i,j) = (1-\rho) d_{\text{sta}}^{k,l}(i,j) + \rho d_{\text{dyn}}^{k,l}(i,j). \tag{8}$$

During training, we linearly increase ρ from 0 to 0.5 so that early matching uses stable contour geometry, while later matching incorporates model-learned cross-period evolutionary evidence.

Multi-marginal optimal transport. To minimize the total evolutionary cost over all observed periods, DOTM uses multi-marginal OT to jointly match samples across periods. For clarity, fix a category and relabel its L observed periods chronologically as $1, \ldots, L$. Let m_q be the empirical variant distribution at period q (uniform by default), $\mathbf{i} = (i_1, \ldots, i_L)$ a candidate trajectory, and $\mathcal{U}(m_1, \ldots, m_L)$ the set of nonnegative joint couplings with these marginals. DOTM estimates the complete-trajectory distribution

$$\Gamma^\star = \underset{\Gamma \in \mathcal{U}(m_1, \ldots, m_L)}{\arg\min} \langle \Gamma, C \rangle + \eta \mathcal{H}(\Gamma), \tag{9}$$

where C is the trajectory-cost tensor with entries $C(\mathbf{i})$, $\Gamma^\star(\mathbf{i})$ assigns probability mass to trajectory $\mathbf{i}$, and $\mathcal{H}$ is negative entropy, encouraging diverse trajectories. Along $\mathbf{i}$, let d_q be the fused matching cost given by Eq. 8 for its selected samples at periods q and $q+1$, and let $\mathbf{s}_q$ be the normalized path-signature static feature of the selected sample at period q. We define the morphological displacement in this feature space as $\mathbf{\Delta}_q = \mathbf{s}_{q+1} - \mathbf{s}_q$. The trajectory cost is

$$C(\mathbf{i}) = \sum_{q=1}^{L-1} d_q + \lambda \sum_{q=2}^{L-1} \|\mathbf{\Delta}_q - \mathbf{\Delta}_{q-1}\|_2^2, \tag{10}$$

where the first term represents the cumulative matching cost between selected variants at adjacent periods, while the second measures the consistency of successive morphological displacements. We compute $\Gamma^\star$ with Sinkhorn belief propagation over overlapping three-period factors, without forming the full coupling tensor (See Appendix E.1 for details).

Use in training. As training proceeds, the EMA copy tracks the evolving network. Every five epochs, we refresh the dynamic features and recompute the joint coupling probability mass, which remains fixed and is used to sample complete chronological trajectories until the next refresh.

3.4 Multi-Path Decoding

Which intermediate periods help the most is unknown at test time and varies across characters. Because placeholders and random period dropping expose the model to many period subsets during training (Sec. 3.2), the same model can decode through different subsets and aggregate their predictions, marginalizing over the choice of intermediate periods.

Let $\mathcal{P}_{\text{mid}} = \{p_2, p_3, p_4, p_5\}$ denote BI, WSC, SS, and CS. Each $S \subseteq \mathcal{P}_{\text{mid}}$ defines a path where the diffusion head generates the selected periods chronologically before RS, while the rest remain as placeholders. Generated glyphs are re-encoded to update the hidden states for later predictions.

Let $\mathcal{S}_{\text{eval}} \subseteq 2^{\mathcal{P}_{\text{mid}}}$ be the paths evaluated at inference. We use all $2^4 = 16$ paths by default, including $S = \varnothing$ for direct OBS→RS decoding. For each $S \in \mathcal{S}_{\text{eval}}$, an OCR classifier maps the final RS glyph $\hat{X}_S^K$ to a top-1 label $\hat{y}_S$ with confidence s_S. We aggregate these path-specific predictions using

$$V(y) = \sum_{S \in \mathcal{S}_{\text{eval}}} \mathbf{1}[\hat{y}_S = y], \qquad Q(y) = \sum_{S \in \mathcal{S}_{\text{eval}}} s_S \mathbf{1}[\hat{y}_S = y]. \tag{11}$$

The final label $\hat{y}$ first maximizes the number of agreeing paths $V(y)$, with summed OCR confidence $Q(y)$ used only to break ties. This majority-first rule prevents a single high-confidence path from overriding broader path agreement. Table 2 compares period pools and fixed-path seed voting under matched final-RS output or glyph-generation budgets. For visualization, we report the path with the highest confidence for the predicted final label $\hat{y}$. We do not claim that the periods highlighted in this representative path are the only ones that contribute to the prediction; for undeciphered characters, it is only an inspectable model hypothesis and remains subject to expert validation.

4 Experiments

4.1 Experimental Setup

We construct a six-period corpus from HUST-OBC (Wang et al., 2024b) and EVOBC (Guan et al., 2024a). The OBS portion contains 1,588 deciphered categories and 72,487 glyph samples; Appendix B reports the category and sample counts for every period in the complete dataset. The

1,588 OBS categories are partitioned into known and held-out sets at a 9:1 ratio. We use three non-overlapping category-level splits and report the average of three complete runs. *Validation* measures unseen samples from known categories; *Decipherment* evaluates held-out deciphered categories excluded from all training, matching, and supervision, whose established paleographic correspondences serve as evaluation ground truth. Outputs for genuinely undeciphered characters are treated as candidate hypotheses for expert assessment (Appendix M).

We train the model for 1000 epochs on four NVIDIA RTX PRO 6000 GPUs. At inference, the four intermediate periods define 16 fixed paths, and multi-path decoding determines the final result. Generated RS glyphs are scored by the in-house recognizer described in Appendix C. For results reported with ± in Tables 1 and 4, values are mean ± s.d. over the three complete split-specific runs. Full dataset, implementation, and evaluation details are provided in Appendices B and C.

4.2 Main Results

Table 1: Overall Top-1 accuracy. Val. measures unseen samples of known categories; Deciph. uses held-out categories. The upper block lists existing methods (underlined: best among them); the lower block trains the ChronoOBS architecture under the paradigms of Fig. 1; Joint is the full model.

Method	Val.	Deciph.	Method	Val.	Deciph.
Classification-based			*VLM-based*		
InceptionV3 (Szegedy et al., 2016)	74.4	–	Gemini 3 Pro (Google DeepMind, 2025)	7.1	5.8
ViT (Dosovitskiy et al., 2021)	79.2	–	GPT-5 (OpenAI, 2025)	8.5	6.7
PyGT (Gan et al., 2023)	<u>84.3</u>	–	PD-OBS (Peng et al., 2025)	80.6	16.8
Direct generation					
BBDM (Li et al., 2023)	55.8	8.2	OBSD (Guan et al., 2024b)	69.4	<u>20.5</u>
Cascaded generation			*Composition-based*		
Sundial-GAN (Chang et al., 2022)	44.3	6.6	P^3 (Wang et al., 2024a)	76.8	13.6
ChronoOBS variants					
Direct (1 seed)	82.6 ± 0.45	21.3 ± 0.27	Cascade	50.2 ± 4.30	9.8 ± 1.48
Direct (16-seed voting)	84.6 ± 0.26	21.8 ± 0.23	Cascade (parameter shared)	33.5 ± 3.64	7.0 ± 1.73
Direct + auxiliary (16-seed voting)	84.8 ± 0.31	22.1 ± 0.25	**Joint**	**86.7** ± 0.16	**24.4** ± 0.20

We compare closed-set classifiers, the composition-based P^3, VLMs, direct generators, and cascaded evolution models. All trainable image-generation baselines use the same category split. Closed-set classifiers receive no decipherment score because held-out categories are outside their label space. The VLM entries use a different prompting and output protocol and are included only as external reference points, with full evaluation details provided in Appendix C.1.

To match the 16 final RS outputs of ChronoOBS's multi-path decoding, each competing image-generation model produces 16 outputs using distinct random seeds and applies the same majority-first voting rule. Table 1 shows that ChronoOBS achieves the strongest performance among the methods evaluated under this protocol. Compared with OBSD, the strongest directly comparable prior image-generation baseline, ChronoOBS improves validation from 69.4% to 86.7% and held-out decipherment from 20.5% to 24.4%. The lower block of Table 1 compares alternative training paradigms using the same underlying ChronoOBS architecture. Because the full Joint result also uses multi-path decoding, Table 2 separately analyzes the contributions of intermediate conditioning, repeated stochastic sampling, and path diversity.

Direct + auxiliary retains direct OBS-to-RS prediction as the main task while adding auxiliary losses for direct OBS-to-intermediate-period generation. Under the same 16-seed direct-decoding protocol, it improves only slightly over Direct, from 84.6%/21.8% to 84.8%/22.1% on validation/decipherment, indicating that auxiliary supervision alone does not account for the performance of the full model. Cascade trains separate adjacent-period translators, whereas Cascade (parameter shared) reuses one model across transitions; both remain restricted to local adjacent-period conditioning and supervision and substantially underperform Direct. Their generated trajectories are also much less identity-consistent: only 9.7% of Cascade trajectories retain the correct character within the verifier's Top-5 at all four intermediate periods (Table 3). Parameter sharing alone therefore does not explain the gap between cascading and Joint. However, the two paradigms also differ in history access, learning

from incomplete lineages, cross-period matching, and inference, so we do not attribute the entire performance gap to any single factor. The controlled comparisons in Table 2 further show that intermediate conditioning and path diversity contribute beyond repeated stochastic sampling.

Table 2 disentangles intermediate-stage use, stochastic ensembling, and path diversity under matched final-RS output budgets and, in a separate control, matched total glyph-generation counts. Its zero-period rows use Joint's direct path, distinct from the separately trained Direct baseline. For a pool of m intermediate periods, the corresponding sub-paths and seeds yield 16 final RS outputs. At this fixed output budget, decipherment rises from 21.5% to 24.4% as the pool expands from zero to four periods. Even without voting, the all-period trajectory reaches 22.7% versus 21.0% for the direct path, consistent with a benefit from intermediate conditioning.

Table 2: Intermediate-period and fixed-path controls. All rows use the same Joint model. Values are mean ± s.d. over all evaluated split-period-pool combinations.

Active periods	Paths × Seeds	Val. ↑	Deciph. ↑
0	1×1	82.3 ± 0.52	21.0 ± 0.29
0	1×16	84.4 ± 0.29	21.5 ± 0.21
1	2×8	85.6 ± 0.38	22.9 ± 0.26
2	4×4	86.1 ± 0.30	23.5 ± 0.19
3	8×2	86.5 ± 0.19	24.0 ± 0.21
4	16×1	**86.7** ± 0.16	**24.4** ± 0.20
Additional controls (one fixed path)			
0	1×48	85.3 ± 0.17	21.9 ± 0.18
4	1×1	83.6 ± 0.23	22.7 ± 0.27
4	1×16	85.9 ± 0.24	23.4 ± 0.24

Under matched glyph-generation compute, 48 direct samples reach 21.9% decipherment accuracy, versus 24.4% from 16 distinct evolutionary paths. Under the same 16-output budget, 16 samples along one fixed all-period path reach 23.4%. These controls support a contribution from path diversity beyond additional diffusion sampling and ordinary stochastic ensembling, although residual differences between training paradigms may remain.

4.3 Trajectory Verification on Known Classes

We evaluate trajectory fidelity only on validation images of known character categories for which period-specific reference glyphs are available. The test measures whether generated forms match their intended script period and preserve character identity across the modeled trajectory. For independent evaluation, DINOv2-based verifiers (Oquab et al., 2024) are trained on image-disjoint real glyphs from known categories. For held-out deciphered or genuinely undeciphered categories, we do not perform quantitative evaluation. It is worth noting that these trajectories should be regarded as model-generated hypotheses, whose validity still requires expert verification.

Table 3: Period and character consistency of generated intermediate trajectories. Real glyphs (GT) provide the verifier reference, and the all-period metric is satisfied only if the ground-truth category appears in the verifier's Top-5 predictions at all four intermediate periods.

	Period	Character Top-5 accuracy					Character retrieval			Trajectory
Setting	Acc.	BI	WSC	SS	CS	Avg.	R@1	R@10	mAP	All-period
Real glyphs (GT)	99.8	93.9	95.3	93.9	97.0	95.0	59.5	92.0	70.4	85.5
Cascade	75.4	56.9	58.1	50.6	48.9	53.6	12.4	18.1	14.7	9.7
ChronoOBS w/o DOTM	90.3	61.3	68.5	78.8	83.6	73.1	52.9	67.7	58.6	59.2
ChronoOBS	**92.6**	**65.7**	**71.1**	**81.2**	**85.4**	**75.8**	**54.3**	**68.2**	**59.2**	**62.4**

Table 3 shows that, on known categories, ChronoOBS produces substantially more recognizable and identity-consistent intermediate trajectories than the cascade: period accuracy rises from 75.4% to 92.6%, average character Top-5 accuracy from 53.6% to 75.8%, and the stricter all-period accuracy from 9.7% to 62.4%. This trajectory-level gap shows that independently trained local mappings preserve character identity less consistently across periods. DOTM further raises period accuracy from 90.3% to 92.6% and all-period accuracy from 59.2% to 62.4%, suggesting that adaptive cross-period matching is associated with reduced drift. Because instance-level historical correspondences are unavailable, this evaluation measures trajectory recognizability and identity preservation along the modeled evolution process, with historical alignment requiring further expert validation.



Figure 3: Qualitative comparison of Direct, Cascade, and Joint generation paradigms.

Figure 3 illustrates these differences: Direct predicts incorrect RS characters, while Cascade drifts in character identity across periods. Joint recovers the correct RS forms in all three examples, with intermediate glyphs consistent with available references despite missing historical records.

4.4 Ablation Studies

Table 4: DOTM ablation against random and static OT matching strategies.

Matching strategy	Val.	Deciph.
Random matching	85.6 ± 0.22	22.9 ± 0.19
Static OT (VGG features)	85.7 ± 0.18	23.1 ± 0.17
Static OT (pixel-space L2)	85.6 ± 0.20	23.2 ± 0.18
Static OT (path signatures)	86.1 ± 0.21	23.7 ± 0.14
DOTM w/o dir. consistency	86.4 ± 0.15	24.1 ± 0.18
Full DOTM	**86.7** ± 0.16	**24.4** ± 0.20

Table 5: Decipherment accuracy as the ratio of missing intermediate periods increases.

	Missing ratio			
Method	0%	25%	50%	75%
Direct	21.8	21.8	21.8	21.8
Cascade	9.8	7.4	3.6	1.6
Direct + auxiliary	22.1	22.1	21.9	21.8
ChronoOBS	**24.4**	**24.0**	**23.3**	**22.2**

All experiments in Table 4 use joint modeling and 16-path multi-path decoding, varying only the matching strategy. Static path-signature OT raises decipherment from 22.9% with random matching to 23.7%, outperforming both VGG-feature and pixel-space L2 matching; adding dynamic distances raises the point estimate to 24.1%, and directional consistency to 24.4%. These results support the value of more aligned intermediate trajectories for modeling joint evolution and the additional contributions of learned representations and cross-period directional consistency.

Table 5 evaluates missing history by randomly dropping intermediate-period samples during training at the indicated ratios; 0% means no additional dropping beyond the dataset's existing missing observations. Cascade collapses from 9.8 to 1.6 at a 75% training drop ratio, reflecting its reliance on an intact adjacent chain. In contrast, ChronoOBS drops from 24.4 to 22.2 and remains best throughout. Its shrinking margin shows that usable historical evidence remains important.

5 Conclusion

We presented ChronoOBS, which organizes chronologically ordered script evidence into a jointly modeled trajectory for Oracle Bone Script decipherment. Specifically, ChronoOBS jointly models all script periods with a shared autoregressive diffusion model, represents missing periods with placeholders, and uses Dynamic Optimal Transport Matching to align unpaired variants. Across controlled comparisons, intermediate historical evidence is most effective when integrated within the joint model and aggregated across alternative inference paths, allowing it to contribute beyond auxiliary targets and chained local translations. ChronoOBS improves held-out decipherment relative to direct and cascaded baselines; on known character categories with period-specific references, its generated trajectories also preserve character identity more consistently. In the future, we will examine the model's hypotheses for undeciphered characters together with paleographers. We will also explore this formulation in other domains with partial and unpaired intermediate records, such as other ancient scripts, historical languages, and biological evolution.

AI USE STATEMENT

In preparing this work, we used generative AI tools to assist with polishing the manuscript, literature research, and parts of the code implementation. The authors have reviewed all AI-assisted content, verified the accuracy and relevance of the cited references, and tested the AI-assisted code for correctness. We take full responsibility for the accuracy, originality, and integrity of the final manuscript and associated artifacts.

REPRODUCIBILITY STATEMENT

The model formulation, training objective, and decoding procedure are described in Secs. 3.1–3.4. Dataset sources, period coverage, and the category-level splitting protocol are documented in Appendix B. Appendix C specifies preprocessing, architecture, optimization settings, computational resources, and the evaluation protocol. DOTM hyperparameters and the matching and trajectory-sampling algorithms are detailed in Appendix E. Additional evaluation protocols, including the alternative OCR evaluator and independent trajectory verification, are provided in Appendices D and J. As described in Sec. 4.1, results are averaged over three complete training and evaluation runs with non-overlapping held-out category sets. The code and trained models will be made publicly available upon acceptance of the paper.

REFERENCES

Yannis Assael, Thea Sommerschield, and Jonathan Prag. Restoring ancient text using deep learning: a case study on Greek epigraphy. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6368–6375. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1668.

Yannis Assael, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603:280–283, 2022. doi: 10.1038/s41586-022-04448-z.

Xiang Chang, Fei Chao, Changjing Shang, and Qiang Shen. Sundial-gan: A cascade generative adversarial networks framework for deciphering oracle bone inscriptions. In *Proc. ACM Int. Conf. Multimedia*, pp. 1195–1203, 2022.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Proc. Advances in Neural Inf. Process. Syst.*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ji Gan, Yuyan Chen, Bo Hu, Jiaxu Leng, Weiqiang Wang, and Xinbo Gao. Characters as graphs: Interpretable handwritten chinese character recognition via pyramid graph transformer. *Pattern Recogn.*, 137:109317, 2023. doi: 10.1016/j.patcog.2023.109317.

Google DeepMind. Gemini 3 Pro model card. Technical report, Google DeepMind, November 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf.

Benjamin Graham. Sparse arrays of signatures for online character recognition. *arXiv preprint arXiv:1308.0371*, 2013. doi: 10.48550/arXiv.1308.0371.

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. *arXiv preprint arXiv:2401.12467*, 2024a.

Haisu Guan, Huanxin Yang, Xinyu Wang, Shengwei Han, Yongge Liu, Lianwen Jin, Xiang Bai, and Yuliang Liu. Deciphering oracle bone language with diffusion models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15554–15567, 2024b.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Moruo Guo and Houxuan Hu (eds.). *Jiaguwen heji*. Zhonghua Shuju, Beijing, 1978.

Isabel Haasler, Rahul Singh, Qinsheng Zhang, Johan Karlsson, and Yongxin Chen. Multi-marginal optimal transport and probabilistic graphical models. *IEEE Transactions on Information Theory*, 67(7):4647–4668, 2021. doi: 10.1109/TIT.2021.3077465.

Wenhui Han, Xinlin Ren, Hangyu Lin, Yanwei Fu, and Xiangyang Xue. Self-supervised learning of orc-bert augmentator for recognizing few-shot oracle characters. In *Proc. Asian Conf. Comp. Vis.*, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proc. Advances in Neural Inf. Process. Syst.*, 33:6840–6851, 2020.

Shuangping Huang, Haobin Wang, Yongge Liu, Xiaosong Shi, and Lianwen Jin. Obc306: A large-scale oracle bone character recognition dataset. In *Proc. Int. Conf. Doc. Anal. and Recognit.*, pp. 681–688. IEEE, 2019.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 1125–1134, 2017.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *Proc. Int. Conf. Mach. Learn.*, pp. 5468–5479, 2020.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *Int. J. Comput. Vis.*, 128:2402–2417, 2020.

Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 1952–1961, 2023.

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*, pp. 19893–19902, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Proc. Advances in Neural Inf. Process. Syst.*, 2024.

Yuanhui Lin, Hetao Wu, Qingju Jiao, Yongge Liu, and Da-Han Wang. Decipherment of oracle bone inscription via component deconstruction and alignment. In *Document Analysis and Recognition – ICDAR 2026*, volume 16972 of *Lecture Notes in Computer Science*, pp. 485–501. Springer, 2026. doi: 10.1007/978-3-032-36023-6_28.

Jiaming Luo, Yuan Cao, and Regina Barzilay. Neural decipherment via minimum-cost flow: From Ugaritic to Linear B. In *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 3146–3155, 2019.

OpenAI. GPT-5 system card. Technical report, OpenAI, 2025. URL https://openai.com/index/gpt-5-system-card/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Kaixin Peng, Mengyang Zhao, Haiyang Yu, Teng Fu, and Bin Li. Interpretable oracle bone script decipherment through radical and pictographic analysis with LVLMs. *arXiv preprint arXiv:2508.10113*, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Jiapeng Wang, Yifan Zhang, Zhuoma GongQue, Chong Sun, Yida Xu, Yadong Xue, et al. V-oracle: Making progressive reasoning in deciphering oracle bones for you and me. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20124–20150, 2025.

Xigui Qiu. *Chinese Writing*. Number 4 in Early China Special Monograph Series. Society for the Study of Early China and the Institute of East Asian Studies, University of California, Berkeley, CA, 2000. ISBN 978-1-55729-071-7. Translated by Gilbert L. Mattos and Jerry Norman.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. doi: 10.1023/A:1026543900054.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4713–4726, 2022b.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Thea Sommerschield, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando de Freitas. Machine learning for ancient languages: A survey. *Comput. Linguist.*, pp. 1–44, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 2818–2826, 2016.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. In *Proc. Int. Conf. Mach. Learn.*, pp. 9526–9536, 2020.

Mei Wang and Weihong Deng. A dataset of oracle characters for benchmarking machine learning algorithms. *Scientific Data*, 11(1):87, 2024.

Mengru Wang, Yu Cai, Li Gao, Ruichen Feng, Qingju Jiao, Xiaolin Ma, and Yu Jia. Study on the evolution of chinese characters based on few-shot learning: From oracle bone inscriptions to regular script. *Plos one*, 17(8):e0272974, 2022.

Pengjie Wang, Kaile Zhang, Xinyu Wang, Shengwei Han, Yongge Liu, Lianwen Jin, Xiang Bai, and Yuliang Liu. Puzzle pieces picker: Deciphering ancient chinese characters with radical reconstruction. In *International Conference on Document Analysis and Recognition*, pp. 169–187. Springer, 2024a.

Pengjie Wang, Kaile Zhang, Xinyu Wang, Shengwei Han, Yongge Liu, Jinpeng Wan, Haisu Guan, Zhebin Kuang, Lianwen Jin, Xiang Bai, and Yuliang Liu. An open dataset for oracle bone character recognition and decipherment. *Scientific Data*, 11(1):976, 2024b. doi: 10.1038/s41597-024-03807-x.

Yun Wang, Zhengchen Li, Dandan Liang, Youyou Zhang, Shouyong Pan, and Yuchun Fang. Deciphering oracle bone characters: Pictographic captioning and reasoning for morphological recognition. *Information Processing & Management*, 63(7):104848, 2026. doi: 10.1016/j.ipm.2026.104848.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Ran Wu, Shibin Wang, Xueshan Li, and Dong Liu. A text image dual conditional stable diffusion model for oracle bone inscription decipherment. *npj Heritage Science*, 13(1):453, 2025.

Gechuan Zhang, Dairui Liu, Barry Smyth, and Ruihai Dong. Deciphering ancient chinese oracle bone inscriptions using case-based reasoning. In *International Conference on Case-Based Reasoning*, pp. 309–324. Springer, 2021.

Xinyi Zhao, Siyuan Liu, Yikai Wang, and Yanwei Fu. Ffd augmentor: Towards few-shot oracle character recognition from scratch. In *Proc. Asian Conf. Comp. Vis.*, pp. 1622–1639, 2022.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 2223–2232, 2017.

A LIMITATIONS

This study has several limitations. First, quantitative evaluation remains recognition-based and is performed on held-out categories whose modern counterparts are already established; this setting may be easier than genuinely undeciphered cases. There is no expert validation on genuinely undeciphered OBS characters. Trajectory verification is restricted to known character categories for which period-specific reference glyphs are available. Consequently, generated trajectories for held-out deciphered or genuinely undeciphered categories should be interpreted as model-generated visual hypotheses rather than validated historical evidence. Second, ChronoOBS organizes historical evidence into six script periods to match the available datasets. This chronological scaffold is a tractable abstraction rather than a claim of unique linear descent between successive script traditions, and it compresses continuous, overlapping, regional, and occasionally abrupt historical changes. The directional-consistency term in DOTM should therefore be understood as a morphological smoothness prior rather than a literal constant-rate model of script evolution. Similarly, the Hilbert ordering used for static path-signature features is a deterministic, locality-aware representation heuristic rather than a claim of geometric invariance or historically correct stroke order. The entropy and smoothness coefficients are fixed implementation choices and were not exhaustively optimized. Third, the framework is primarily morphological, whereas authoritative decipherment also depends on inscription-level context, semantics, syntax, textual corroboration, and broader historical knowledge. Moreover, the current decoder always returns a modern character, although some undeciphered forms may have no secure modern counterpart. This forced-choice setting can produce plausible-looking but unsupported hypotheses; an important extension is therefore to support abstention when the available evidence is insufficient for a reliable correspondence. Finally, our empirical validation is limited to OBS. Although the formulation may be relevant to other problems with partial and unpaired intermediate observations, transfer beyond OBS remains to be empirically established.

B DATASET DETAILS

This section complements the dataset description in Sec. 4.1. The complete corpus contains glyphs from all six script periods, with substantially different coverage across the source collections. Table 6 reports the number of character categories and glyph samples available in each period. Quantitative decipherment evaluation is defined on the 1,588 deciphered OBS categories, which contain 72,487 OBS glyph samples and have established modern labels. Genuinely undeciphered OBS categories are excluded from quantitative accuracy evaluation and are used only for qualitative hypothesis generation in Appendix M. Table 7 summarizes the availability of intermediate-period records among the 1,588 evaluated categories.

Table 6: Per-period category and sample counts in the complete dataset.

Period	Categories	Samples
OBS	1,588	72,487
BI	3,452	47,314
WSC	7,222	80,042
SS	9,147	13,434
CS	2,890	3,568
RS	88,899	88,899

Table 7: Category coverage by intermediate-period count; shares are rounded to two decimals.

Intermediate periods	Categories	Share
0	103	6.49%
1	265	16.69%
2	300	18.89%
3	315	19.84%
4 (complete)	**605**	**38.10%**
Total	1,588	100%

The 1,588 deciphered OBS categories are divided into known and held-out categories at a 9:1 ratio. We construct three category-level splits with non-overlapping held-out category sets and repeat the complete training and evaluation procedure for each split. For validation, glyph samples from known categories are further divided into training and validation subsets at a 9:1 ratio. For decipherment, all glyphs belonging to a held-out category across all script periods are excluded from model training, DOTM matching, and generative supervision; the established modern label is used only after generation for evaluation. Thus, the decipherment task measures category-level generalization rather than recognition of additional samples from categories observed during training.

C Implementation Details

All glyph images are converted to a unified foreground–background convention and resized to 96×96. We retain every available glyph variant at each period so that DOTM can model many-to-many correspondence. A missing position uses its learnable period embedding as the placeholder token and receives no period-specific generative supervision. During training, each available intermediate glyph is also independently dropped from the conditioning sequence with probability 0.5; this augmentation changes the conditioning history but not which available target periods receive supervision. The image encoder outputs 768-dimensional glyph features, which are combined with period embeddings before entering a 12-layer, 12-head autoregressive transformer of hidden width 768. Its lower-triangular chronological mask restricts the hidden state at p_k to positions $p_1, \ldots, p_k$. All post-OBS periods share one lightweight DiT head, which is conditioned only on the preceding evolutionary hidden state h_c^{k-1}; period embeddings enter the autoregressive transformer but are not supplied directly to the DiT head.

For LSS, each noisy target glyph and its projected condition map are cropped into overlapping 64×64 patches with stride 16. Each optimizer batch supplies eight randomly located local patches for generative supervision. At inference, predictions from all local crops are pasted back onto the full canvas and averaged in overlapping regions, yielding one complete glyph at every ODE integration step.

The model is trained with Adam for 1000 epochs using a learning rate of 2×10^{-5}, weight decay 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a batch size of 8. Training uses four NVIDIA RTX PRO 6000 GPUs and takes approximately one week. At inference, OBS and RS are always active, while BI, WSC, SS, and CS define $2^4 = 16$ period-subset paths. Each selected period is generated autoregressively with 50 numerical ODE integration steps. After an intermediate glyph $\hat{X}_S^k$ is generated, its token is updated to $E_{\text{img}}(\hat{X}_S^k) + e_k$ before the transformer recomputes the hidden states for the next selected period; skipped positions retain their placeholder tokens.

For all image-generation methods, generated RS glyphs are mapped to ranked modern-character predictions using an in-house ResNet-101 recognizer trained over 88,899 Regular Script categories. We refer to this evaluator as RS-OCR. It achieves 99.87% accuracy on clean RS glyphs. A generated prediction is counted as correct at Top-k when the ground-truth modern character occurs among the first k entries of the RS-OCR ranking. These scores measure automatic recognizability rather than paleographic validity: visual artifacts can alter the OCR ranking, and expert validation remains necessary for actual decipherment. Except for the explicitly labeled Direct (1 seed) control, every image-generation method in the main comparison produces 16 final RS outputs and applies the same majority-first voting rule, with summed RS-OCR confidence used only to break ties. Thus, the main comparison controls the number of final RS outputs and the post-hoc evaluator, but not the total number of generated intermediate glyphs. Table 2 therefore includes separate fixed-output and matched glyph-generation controls. Appendix K reports the full RS-OCR Top-k comparison, while Appendix D replaces only the post-hoc recognizer with PP-OCRv5 to evaluate sensitivity to the choice of OCR evaluator.

The cascaded baselines use the same category splits and train each adjacent-period translator only from categories observed at both endpoints of that transition; a missing endpoint provides no pairwise supervision for that step. Sundial-GAN and the Cascade variant of ChronoOBS train five independent translators along the OBS–BI–WSC–SS–CS–RS sequence, whereas Cascade (parameter shared) reuses one translator across transitions. At inference, all cascaded baselines compose adjacent predictions sequentially and use the same 16-output voting and RS-OCR evaluation protocol. These baselines characterize the local-transition paradigm; because they also differ from Joint in source/history access, incomplete-lineage supervision, and matching, the performance gap should not be attributed to parameter sharing alone.

C.1 VLM Baseline Evaluation

The generic VLM baselines in Table 1, Gemini 3 Pro and GPT-5, directly predict modern Chinese characters from OBS glyph images. Following the visual-reasoning setting of Peng et al. (2025), we use different prompt protocols for known-category validation and held-out-category decipherment. Each query produces one response containing at most 10 ranked candidate characters, with external

retrieval and tool use disabled where applicable. These methods are evaluated by direct character matching rather than RS-OCR. Because the VLMs use a prompting-based prediction protocol rather than the 16-output image-generation and voting procedure used by the generative baselines, their scores are included as external reference points rather than as protocol-matched comparisons.

Known-category validation. We provide a small in-context reference set of OBS glyphs and their modern Chinese labels, sampled only from the training split of known categories. The references contain neither validation images nor glyphs from held-out categories. The query is an unseen validation image. We use the following prompt:

> You are given several labeled reference examples of Oracle Bone Script glyphs and their modern Chinese character labels. Use these references to infer the label of the query Oracle Bone Script glyph. Return only a ranked list of at most 10 candidate modern Chinese characters, without explanations.

Held-out-category decipherment. We provide only the query OBS glyph image, without labeled reference examples, and use the following prompt:

> Given this Oracle Bone Script glyph image, infer its most likely modern Chinese character counterpart. Return only a ranked list of at most 10 candidate modern Chinese characters, without explanations.

Response processing and scoring. If a response includes explanations, we extract the returned candidate characters. Duplicate candidates are removed while preserving the order of their first occurrence. For the Top-1 results in Table 1, only the first candidate is compared with the ground-truth modern character; a correct answer appearing later in the list does not count as a Top-1 hit.

D PaddleOCR Evaluation

The primary OCR-based evaluation uses RS-OCR, the in-house ResNet-101 recognizer described in Appendix C. As a complementary evaluator-sensitivity check, we additionally recognize the generated RS glyphs with PP-OCRv5 from the open-source PaddleOCR toolkit.[1] PP-OCRv5 is external to our training pipeline and is not used for training, model selection, or matching.

This analysis retains the category-level splits, generated images, 16-output inference budget, and majority-first aggregation protocol of the main comparison; only the post-hoc recognizer mapping generated RS glyphs to modern characters is changed. Table 8 reports methods for which PP-OCRv5 predictions are available. The RS-OCR columns reproduce the corresponding results in Table 1.

Table 8: Supplementary PP-OCRv5 Top-1 evaluation on the same generated Regular Script outputs. All methods use the 16-output protocol of Table 1; only the post-hoc OCR recognizer changes.

Method	Validation		Decipherment	
	RS-OCR	PP-OCRv5	RS-OCR	PP-OCRv5
Pix2Pix (Isola et al., 2017)	5.3	4.3	0.0	0.0
Palette (Saharia et al., 2022a)	7.9	6.6	0.0	0.0
DRIT++ (Lee et al., 2020)	11.6	9.9	0.1	0.1
CycleGAN (Zhu et al., 2017)	18.4	15.8	0.3	0.2
Sundial-GAN (Chang et al., 2022)	44.3	43.9	6.6	4.8
BBDM (Li et al., 2023)	55.8	48.8	8.2	6.1
CDE (Saharia et al., 2022b)	61.9	54.5	17.4	13.4
OBSD (Guan et al., 2024b)	69.4	61.0	20.5	16.2
ChronoOBS	**86.7**	**76.6**	**24.4**	**19.4**

Absolute accuracies are lower with PP-OCRv5 than with the task-specific RS-OCR, reflecting differences in label coverage, recognition pipeline, and confidence calibration. Nevertheless, the

[1] https://github.com/PaddlePaddle/PaddleOCR

relative ordering is preserved: ChronoOBS remains stronger than the compared image-generation baselines on both validation and held-out-category decipherment. This agreement indicates that the main comparison is not specific to the in-house OCR evaluator. At the same time, the absolute score differences reinforce that OCR-based accuracy measures automatic recognizability rather than paleographic validity.

E Dynamic Optimal Transport Matching Details

DOTM is refreshed every five epochs using path-signature static distance, dynamic feature distance, entropy weight $\eta = 2$, smoothness weight $\lambda = 0.5$, and 20 Sinkhorn belief-propagation iterations. These fixed values specify the implementation used for the reported experiments and are provided for reproducibility rather than as uniquely optimal settings. We use Hilbert ordering as a deterministic, locality-aware heuristic for constructing a one-dimensional representation of the glyph; it is not intended to recover historical stroke order or to provide geometric invariance. For the resulting ordered curve $\gamma_X : [0,1] \to \mathbb{R}^2$ of glyph X, the order-M truncated signature is

$$\begin{aligned} S_M(\gamma_X) &= \{I_m(\gamma_X)\}_{m=0}^{M}, \\ I_m(\gamma_X) &= \int_{0<t_1<\cdots<t_m<1} d\gamma_{X,t_1} \otimes \cdots \otimes d\gamma_{X,t_m}, \end{aligned} \tag{12}$$

where $I_0 = 1$. After relabeling the observed periods as $q = 1, \ldots, L$, let $\mathbf{s}_{q,i}$ be the ℓ_2-normalized signature of variant i. At matching update r, each candidate glyph is placed at its original chronological position with placeholders elsewhere, and the normalized hidden state of the EMA encoder–transformer is denoted by $\bar{\mathbf{h}}_{q,i}^{(r)}$. The distances used in Eq. 8 are therefore

$$\begin{aligned} d_{\mathrm{sta}}^{q}(i,j) &= \|\mathbf{s}_{q,i} - \mathbf{s}_{q+1,j}\|_2^2, \qquad d_{\mathrm{dyn}}^{q,(r)}(i,j) = \|\bar{\mathbf{h}}_{q,i}^{(r)} - \bar{\mathbf{h}}_{q+1,j}^{(r)}\|_2^2, \\ d_{q}^{(r)}(i,j) &= (1-\rho_r) d_{\mathrm{sta}}^{q}(i,j) + \rho_r d_{\mathrm{dyn}}^{q,(r)}(i,j), \end{aligned} \tag{13}$$

where $\rho_r = 0.5(r-1)/(R-1)$ for R matching updates. For incomplete categories, q and $q+1$ denote consecutive available periods and therefore need not be adjacent stages in the canonical six-period ordering. Accordingly, the directional-consistency term treats successive available observations as comparable morphological steps. We use this term as a smoothness prior over the modeled trajectory rather than as a claim that historical script change proceeds at a constant rate or over equal temporal intervals.

Fig. 4 illustrates how these distances connect matching to model training. The purple branch provides a fixed geometric reference from Hilbert-ordered contours, while the blue branch updates the matching evidence as the EMA representations evolve. Their weighted fusion in Eq. 13 supplies the pairwise terms of the trajectory cost; the directional-consistency term additionally links successive transitions. The resulting joint coupling supplies complete training trajectories, closing the loop between representation learning and correspondence refinement.

E.1 Joint Coupling and Trajectory Sampling

Standard OT computes a separate pairwise plan $\Pi^{k,l}$ for each pair of periods; DOTM instead optimizes the single multi-period coupling $\Gamma^{\star}(i_1, \ldots, i_L)$. Suppressing the category and refresh indices, its feasible set is

$$\mathcal{U}(m_1, \ldots, m_L) = \left\{ \Gamma \geq 0 : \sum_{\mathbf{i}: i_q = a} \Gamma(\mathbf{i}) = m_q(a),\ \forall q, a \right\}, \tag{14}$$

and $\mathcal{H}(\Gamma) = \sum_{\mathbf{i}} \Gamma(\mathbf{i})[\log \Gamma(\mathbf{i}) - 1]$ in Eq. 9. We use $m_q(a) = 1/|\mathcal{X}^q|$ unless another prior is available; hence these constraints preserve the empirical variant distribution at every observed period.

For $L \geq 3$, define an overlapping three-period cost for $q = 2, \ldots, L-1$ as

$$\begin{aligned} g_q(i_{q-1}, i_q, i_{q+1}) = a_q d_{q-1}(i_{q-1}, i_q) + b_q d_q(i_q, i_{q+1}) \\ + \lambda \|(\mathbf{s}_{q+1,i_{q+1}} - \mathbf{s}_{q,i_q}) - (\mathbf{s}_{q,i_q} - \mathbf{s}_{q-1,i_{q-1}})\|_2^2, \end{aligned} \tag{15}$$

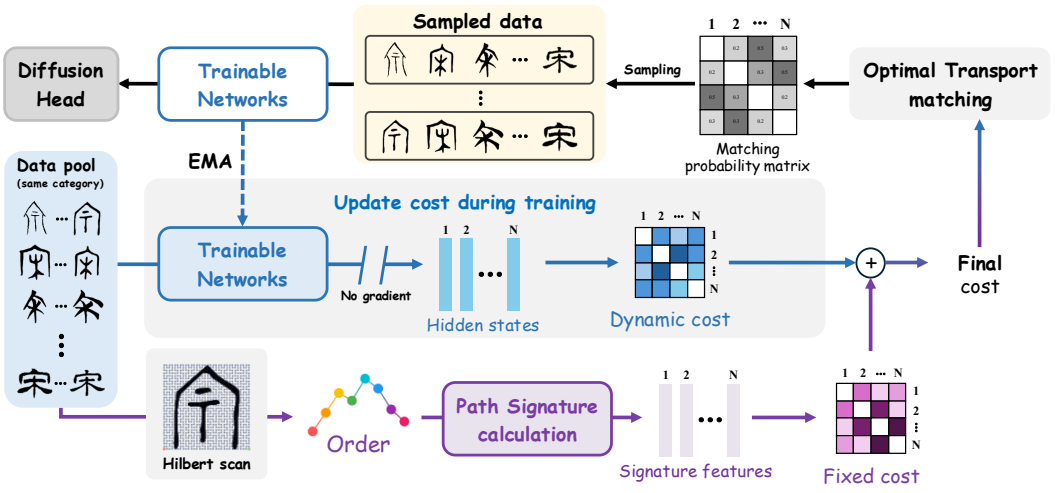


Figure 4: Static–dynamic distance construction and the DOTM training loop. Static path-signature distances (purple) are fused with dynamic EMA-feature distances (blue); the plus symbol denotes the weighted fusion in Eq. 13. Matching updates and trajectory sampling operate outside backpropagation. The displayed probability matrix is schematic: DOTM samples complete trajectories from a multi-period joint coupling, as detailed in Sec. E.1.

where $a_2 = b_{L-1} = 1$ and all other a_q, b_q equal $1/2$. Consequently, every adjacent cost is counted exactly once and $C(\mathbf{i}) = \sum_{q=2}^{L-1} g_q(i_{q-1}, i_q, i_{q+1})$, matching Eq. 10. For $L = 2$, we instead use the pair factor $\psi_1(i_1, i_2) = \exp[-d_1(i_1, i_2)/\eta]$ and no smoothness term.

For $L \geq 3$, let $\psi_q = \exp(-g_q/\eta)$. Given positive scaling vectors u_q, define the implicit joint distribution

$$\Gamma_{\mathbf{u}}(\mathbf{i}) = \left[\prod_{q=1}^{L} u_q(i_q)\right] \left[\prod_{q=2}^{L-1} \psi_q(i_{q-1}, i_q, i_{q+1})\right], \tag{16}$$

where the scalings absorb the normalizing constant. Sinkhorn belief propagation (Cuturi, 2013; Haasler et al., 2021) evaluates its current single-period mass

$$\mu_q(a) = \sum_{\mathbf{i}: i_q = a} \Gamma_{\mathbf{u}}(\mathbf{i}), \qquad u_q(a) \leftarrow u_q(a) \frac{m_q(a)}{\mu_q(a)}, \tag{17}$$

using sum–product messages on the second-order chain. At convergence, the constraints hold and $\Gamma_{\mathbf{u}} = \Gamma^\star$. We count one complete cycle over $q = 1, \ldots, L$ as one Sinkhorn belief-propagation iteration and use 20 iterations. Denote the resulting normalized joint approximation by $\widehat{\Gamma}^{(r)}$; its scaling vectors and messages are retained until the next DOTM refresh, while the L-dimensional tensor is never materialized.

Complete-trajectory sampling is performed from this representation. Define backward messages $B_L(a, b) = 1$ and, for $q = L - 1, \ldots, 2$,

$$B_q(a, b) = \sum_j \psi_q(a, b, j) u_{q+1}(j) B_{q+1}(b, j). \tag{18}$$

For $L \geq 3$, we first draw (i_1, i_2) with probability proportional to $u_1(i_1) u_2(i_2) B_2(i_1, i_2)$ and then draw, for $q = 2, \ldots, L - 1$,

$$\Pr(i_{q+1} = j \mid i_{q-1}, i_q) = \frac{\psi_q(i_{q-1}, i_q, j) u_{q+1}(j) B_{q+1}(i_q, j)}{B_q(i_{q-1}, i_q)}. \tag{19}$$

For $L = 2$, the pair is drawn with probability proportional to $u_1(i_1) u_2(i_2) \psi_1(i_1, i_2)$. Both cases produce one coherent index tuple drawn from the full joint approximation,

$$\mathbf{i} = (i_1, \ldots, i_L) \sim \widehat{\Gamma}^{(r)}, \qquad \widehat{\Gamma}^{(r)} \approx \Gamma^\star, \tag{20}$$

rather than assembling independently selected transitions, and therefore retains the directional-consistency dependence in Eq. 10. The selected indices instantiate the observed glyphs in the training sequence; absent chronological positions are then filled with placeholders. Sampling and the periodic coupling update are outside backpropagation.

Algorithm 1 DOTM update and complete-trajectory sampling

Require: Variant sets and masses $\{\mathcal{X}^q, m_q\}_{q=1}^{L}$; EMA model; η, λ; 20 Sinkhorn-BP iterations

Compute $\mathbf{s}_{q,i}$, $\bar{\mathbf{h}}_{q,i}^{(r)}$, and $d_q^{(r)}$ using Eq. 13.
Construct g_q, ψ_q (or ψ_1 if $L = 2$) and initialize positive scalings u_q.
for $s = 1, \ldots, 20$ **do**
  for $q = 1, \ldots, L$ **do**
    Run sum–product message passing to obtain μ_q and apply Eq. 17.
  end for
end for
if $L \geq 3$ **then**
  Compute Eq. 18 and draw $\mathbf{i}$ using Eq. 19.
else
  Draw (i_1, i_2) from the normalized two-period factor.
end if
Insert the selected glyphs at observed periods and placeholders elsewhere.

Ensure: A chronological training sequence whose observed tuple follows the calibrated joint approximation.

E.2 Correspondence Visualization

Fig. 5 compares cross-period correspondence weights from static path-signature OT and DOTM for the same character category. Rows and columns index glyph variants from two periods, with identical ordering in both panels. The annotated values show how each source variant distributes its correspondence weight across candidate target variants.

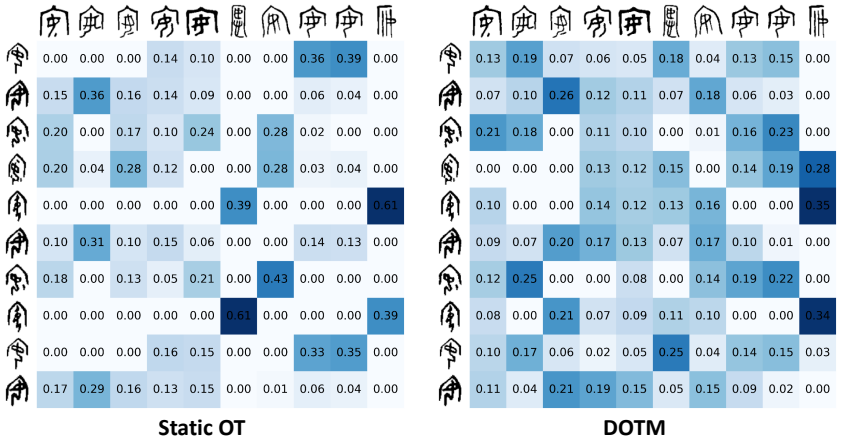


Figure 5: Cross-period correspondence weights under static OT (left) and DOTM (right), using identical glyph ordering. Darker cells indicate larger weights within each panel; annotated values support comparison across panels.

Static OT concentrates several rows on a small set of candidates, whereas DOTM redistributes probability mass over additional alternatives and can change the preferred candidates after incorporating learned dynamic features. For example, rows 5 and 8 become less concentrated under DOTM, and row 3 shifts substantial mass from one candidate to another. These examples illustrate how the adaptive cost changes soft associations used for trajectory sampling; they do not constitute ground-truth validation of individual historical correspondences.

This redistribution is consistent with the learned cross-period features and directional-consistency term in DOTM's matching cost. Sampling complete trajectories from the resulting coupling exposes training to alternative combinations of observed forms (Sec. E.1), while the displayed pairwise weights provide only a local view of the underlying multi-period coupling. Quantitatively, replacing

static path-signature OT with full DOTM raises validation/decipherment from 86.1%/23.7% to 86.7%/24.4% (Table 4). Because instance-level cross-period correspondences are not recorded in the source data, this downstream gain should be interpreted as evidence of the utility of adaptive matching rather than as direct correspondence accuracy.

F Cross-Model Fusion over Fixed Training-Period Configurations

Table 9 removes the training supervision of each intermediate period in turn. Removing any single period lowers both metrics relative to the full model.

Table 9: Leave-one-period-out ablation over the four intermediate periods.

Setting	Val.	Deciph.
ChronoOBS	**86.7**	**24.4**
w/o BI	86.0	<u>23.8</u>
w/o WSC	<u>86.2</u>	23.7
w/o SS	<u>86.2</u>	23.5
w/o CS	86.1	23.5

The leave-one-period-out results in Table 9 use the full multi-path voting decoder. As a complementary diagnostic, Table 10 trains 16 separate models, one for each fixed subset of the four intermediate periods. Each of the first 16 rows evaluates one model only on the path matching its own training-period configuration. The final row then applies multi-path voting across these 16 independently trained model–path pairs. Thus, in this table, voting is used to evaluate *cross-model path fusion*: both the model parameters and the fixed path differ among the predictions being fused. The lower block averages the individual-model results for configurations containing the same number of active intermediate periods; the zero- and four-period groups each contain one configuration.

Table 10: Cross-model fusion over fixed training-period configurations. The 16 configuration rows report 16 separately trained models, each supervised and evaluated only on its checkmarked path. The final row applies multi-path voting to one prediction from each of these 16 independent models, thereby measuring path fusion across models. OBS and RS are always included.

OBS	BI	WSC	SS	CS	RS	Validation	Decipherment
✓					✓	82.6	21.3
✓	✓				✓	84.8	22.0
✓		✓			✓	85.1	21.8
✓			✓		✓	**85.7**	22.8
✓				✓	✓	85.3	22.6
✓	✓	✓			✓	84.9	22.2
✓	✓		✓		✓	85.3	<u>23.3</u>
✓	✓			✓	✓	84.6	22.0
✓		✓	✓		✓	85.1	22.8
✓		✓		✓	✓	84.9	22.5
✓			✓	✓	✓	**85.7**	<u>23.5</u>
✓	✓	✓	✓		✓	84.7	22.7
✓	✓	✓		✓	✓	84.0	22.5
✓	✓		✓	✓	✓	84.6	23.2
✓		✓	✓	✓	✓	84.7	23.0
✓	✓	✓	✓	✓	✓	84.6	**23.6**
Mean accuracy by number of active intermediate periods							
0 period (1 configuration)						82.6	21.3
1 period (4 configurations)						85.2	22.3
2 periods (6 configurations)						85.1	22.7
3 periods (4 configurations)						84.5	22.9
4 periods (1 configuration)						84.6	23.6
Multi-path voting (16 models)						**86.4**	**23.9**

Across the 16 independently trained models, introducing intermediate scripts generally improves decipherment, although the gain is not monotonic because period coverage, sample quality, and matching uncertainty differ across configurations. The highest validation accuracy, 85.7%, is attained by both the SS-only and SS+CS models, whereas the all-period model achieves the highest individual-model decipherment accuracy of 23.6%; SS+CS is close at 23.5%. Fusing one prediction from each of the 16 independently trained model–path pairs yields 86.4%/23.9%, above every constituent configuration, providing evidence that predictions from different period configurations can be complementary. This analysis addresses a different question from Table 11, whose 86.7%/24.4% result fuses 16 inference paths generated by one fixed model.

G Within-Model Fusion over Evolutionary Inference Paths

Table 11 instead keeps a single model fixed: the model is trained once using all six script periods, and only the active intermediate-period subset is changed at inference. Its 16 rows are therefore 16 evolutionary inference paths of the same model, not 16 separately trained models. The final row applies multi-path voting across these within-model paths to evaluate whether alternative evolutionary rollouts from shared parameters provide complementary evidence.

Table 11: Within-model fusion over evolutionary inference paths. All 16 rows use the same model trained on all six script periods; each row activates a different subset of intermediate periods only at inference. The final row applies multi-path voting across these 16 paths, thereby measuring path fusion within one fixed model.

OBS	BI	WSC	SS	CS	RS	Validation	Decipherment
✓					✓	82.3	21.0
✓	✓				✓	83.3	21.6
✓		✓			✓	83.4	21.4
✓			✓		✓	**85.5**	22.9
✓				✓	✓	83.9	22.3
✓	✓	✓			✓	83.5	21.9
✓	✓		✓		✓	84.3	23.1
✓	✓			✓	✓	83.6	21.8
✓		✓	✓		✓	85.3	23.0
✓		✓		✓	✓	84.4	22.5
✓			✓	✓	✓	**85.5**	**23.6**
✓	✓	✓	✓		✓	84.1	22.7
✓	✓	✓		✓	✓	83.2	22.4
✓	✓		✓	✓	✓	84.1	23.2
✓		✓	✓	✓	✓	85.0	23.3
✓	✓	✓	✓	✓	✓	83.6	22.7
Multi-path voting (one model)						**86.7**	**24.4**

The individual rows in Table 11 evaluate each chronological rollout directly, without aggregation, while keeping the trained parameters identical. They diagnose path-specific behavior; the final row is the default within-model voting result. In contrast, Table 10 changes the trained model together with the period configuration and uses voting to measure fusion across 16 independent models.

Importantly, chronological ordering does not imply that every intermediate period is informative for every input. The longest fixed path is not the strongest individual rollout: SS+CS reaches 85.5% validation and 23.6% decipherment, whereas the all-period rollout reaches 83.6% and 22.7%. Performance is therefore not monotonic in path length, reflecting differences in period coverage, sample quality, and the usefulness of particular intermediate bridges. No individual path from this fixed model reaches the final 86.7%/24.4% result; aggregating predictions across all 16 period subsets yields the highest accuracy. Together with the fixed-path controls in Table 2, this supports a contribution from inference-path diversity.

H Restricted Evolution-Path Pool Analysis

Table 12 reports the best independently selected pool at each size and summarizes period usage in the representative paths selected by the full decoder, complementing the per-size averages in Table 2 and Table 11. Pools at successive sizes need not be nested.

Table 12: Period usage in selected paths and accuracy of the best independently selected pool at each size.

Analysis	Setting	Value	Validation	Decipherment
Selected-path period usage				
Frequency	BI	50.27%	–	–
Frequency	WSC	50.51%	–	–
Frequency	SS	53.31%	–	–
Frequency	CS	51.21%	–	–
Best available-period pool at each size				
Pool size 0	none (OBS→RS)	1 path	82.3	21.0
Pool size 1	SS	2 paths	85.8	23.2
Pool size 2	WSC+SS	4 paths	86.5	23.7
Pool size 3	BI+SS+CS	8 paths	**86.7**	24.1
Pool size 4	BI+WSC+SS+CS	16 paths	**86.7**	**24.4**

Table 12 and Fig. 6 provide a descriptive view of the representative paths returned by the decoder. Each intermediate period appears in roughly half of these paths. Because uniform use of all 16 period subsets would also yield a 50% marginal inclusion rate for every period, these frequencies should not be interpreted as evidence that individual periods are preferentially selected or uniquely responsible for particular predictions. Evidence for complementary inference paths instead comes from the fixed-path and repeated-sampling controls in Table 2. The independently evaluated period pools show that the best one-period pool (SS) reaches 85.8%/23.2%, the best two-period pool (WSC+SS) reaches 86.5%/23.7%, and the best three-period pool (BI+SS+CS) reaches 86.7%/24.1%. The full four-period pool reaches 86.7%/24.4%. Thus, performance largely saturates after three available intermediate periods on validation, while the complete 16-path pool provides a smaller additional gain on held-out decipherment.

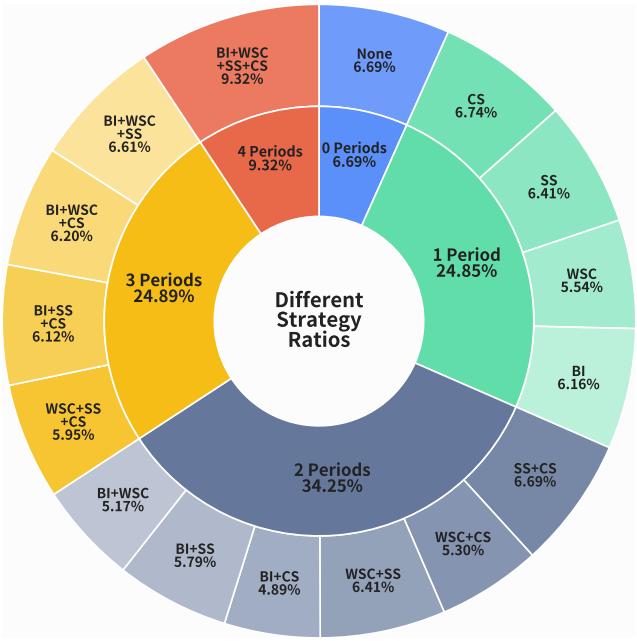


Figure 6: Representative paths selected by multi-path voting. Inner: active-period count; outer: period subsets.

I REPEATED-SAMPLING DIAGNOSTIC

Table 13: Accumulated hit rate under repeated random-seed sampling on the fixed all-period path. An input is counted as correct if any accumulated RS-OCR Top-1 prediction matches the target.

Metric / Trial	1	2	3	4	5
Validation	83.6%	85.8%	86.5%	86.9%	87.1%
Decipherment	22.7%	29.1%	33.4%	36.3%	38.4%
Metric / Trial	**6**	**7**	**8**	**9**	**10**
Validation	87.3%	87.6%	87.7%	87.9%	87.9%
Decipherment	39.8%	40.6%	41.0%	41.1%	41.2%

Unlike the 16-seed majority-vote control in Table 2, this is an accumulated-candidate, recall-like diagnostic: a sample is counted as correct once any accumulated seed produces an RS-OCR Top-1 prediction matching the target. The accumulated hit rate rises from 83.6% to 87.9% on validation and from 22.7% to 41.2% on decipherment over ten trials. The larger increase for held-out categories shows that stochastic generation can expose a correct candidate even when the first sample is incorrect. However, this oracle-style diagnostic asks only whether a correct candidate has appeared and does not identify that candidate automatically. The gap between accumulated hit rate and majority-vote accuracy therefore separates candidate generation from reliable candidate selection, which remains a central difficulty in zero-shot decipherment.

J ADDITIONAL TRAJECTORY VERIFICATION

Trajectory verification is restricted to validation data from known character categories, where reliable period-specific reference glyphs and retrieval galleries are available. We use post-hoc DINOv2-based verifiers trained exclusively on authentic, image-disjoint glyphs from known categories to recognize the historical period and character category of generated intermediate glyphs. Their training data exclude the validation glyphs and all generated glyphs. Held-out deciphered categories appear in neither the verifier label spaces nor the retrieval galleries. The verifiers are frozen during evaluation and never provide gradients, losses, labels, or retrieval signals to ChronoOBS. Accordingly, the metrics in this section measure trajectory fidelity only for known categories and should not be interpreted as quantitative validation of trajectories for held-out deciphered or genuinely undeciphered characters.

Table 14: Period-wise independent verification under the all-period rollout. Overall is sample-pooled across BI, WSC, SS, and CS. Cls. denotes period-conditioned character classification.

Period	Period Top-1	Cls. Top-1	Cls. Top-5	R@1	R@5	R@10	mAP
BI	84.7	43.2	65.7	40.8	53.2	59.3	46.9
WSC	99.7	49.2	71.1	49.5	61.2	65.3	55.2
SS	91.4	67.7	81.2	59.9	68.8	71.6	64.3
CS	94.6	74.0	85.4	68.6	75.0	77.5	71.7
Overall	92.6	58.5	75.8	54.3	64.3	68.2	59.2

The period-wise breakdown complements the aggregate and trajectory-consistency results in Table 3. Period recognition is strong throughout, reaching 92.6% overall and 99.7% for WSC, whereas period-conditioned character Top-1 is 58.5% overall. This gap indicates that the generator usually captures the style of the intended historical period before it fully preserves category-specific structure. BI is the most difficult stage for identity preservation, with 43.2% character Top-1, 40.8% Recall@1, and 46.9% mAP; all three measures improve toward SS and CS. Category drift in the early, highly variable script periods is the primary remaining trajectory error.

Complementing Sec. 4.3, incorporating more periods consistently improves retrieval performance: Recall@1 increases from 51.5% to 54.3%, Recall@10 from 65.2% to 68.2%, and mAP from 56.3% to 59.2%. In contrast, character-classification Top-1 changes only from 57.5% to 58.5%. Additional intermediate-period conditioning primarily improves the organization of generated trajectories in

Table 15: Trajectory verification averaged over decoding paths with the same number of active intermediate periods. Cls. Top-1 denotes the independent verifier's character-classification accuracy.

Active periods	Period Top-1	Cls. Top-1	R@1	R@5	R@10	mAP
1	91.6	57.5	51.5	61.2	65.2	56.3
2	92.5	57.8	52.6	62.3	66.3	57.4
3	92.6	58.5	53.5	63.4	67.4	58.4
4	92.6	58.5	54.3	64.3	68.2	59.2

the verifier's feature space, with a smaller change in classification at individual stages. This pattern supports the role of intermediate periods as cumulative structural constraints.

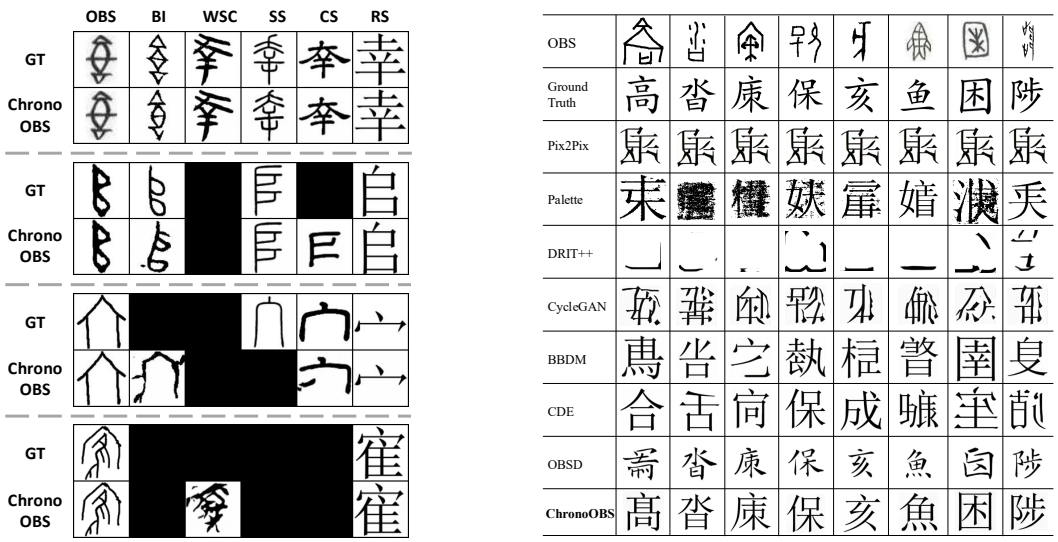
K Generative Decipherment Evaluation

To examine ranking quality beyond Top-1, we compare ChronoOBS with image-to-image translation baselines adapted to OBS decipherment. The direct GAN-based group includes Pix2Pix (Isola et al., 2017), CycleGAN (Zhu et al., 2017), and DRIT++ (Lee et al., 2020); Sundial-GAN (Chang et al., 2022) represents cascaded multi-period translation; and the diffusion-based group includes BBDM (Li et al., 2023), CDE (Saharia et al., 2022b), Palette (Saharia et al., 2022a), and OBSD (Guan et al., 2024b). All methods in this table use the same category splits, RS-OCR evaluator, 16-output inference budget, and majority-first aggregation protocol. After aggregation, Top-k accuracy is computed from the associated RS-OCR ranking. The Top-1 rows in Table 16 therefore reproduce the corresponding image-generation results in Table 1; Pix2Pix, CycleGAN, DRIT++, CDE, and Palette are reported only in this appendix.

Table 16: Top-k accuracy of generative methods using the same RS-OCR evaluator and 16-output protocol as the main comparison.

Rank	Pix2Pix	Palette	DRIT++	CycleGAN	Sundial-GAN	BBDM	CDE	OBSD	ChronoOBS
Validation									
Top-1	5.3%	7.9%	11.6%	18.4%	44.3%	55.8%	61.9%	69.4%	**86.7%**
Top-10	12.0%	16.2%	22.1%	31.8%	48.8%	66.5%	73.4%	78.3%	**87.4%**
Top-20	17.8%	23.7%	30.5%	42.1%	50.2%	70.1%	77.0%	81.5%	**87.8%**
Top-50	28.1%	36.0%	41.9%	55.6%	52.6%	74.8%	81.6%	85.0%	**88.5%**
Top-100	38.5%	47.8%	55.2%	68.3%	54.2%	79.2%	85.5%	88.4%	**89.1%**
Top-200	39.2%	49.3%	56.0%	69.5%	54.9%	80.0%	86.1%	88.5%	**89.4%**
Top-500	39.6%	49.8%	56.7%	70.0%	55.2%	80.4%	86.3%	88.7%	**89.9%**
Decipherment									
Top-1	0.0%	0.0%	0.1%	0.3%	6.6%	8.2%	17.4%	20.5%	**24.4%**
Top-10	0.1%	0.2%	0.6%	1.1%	9.5%	14.7%	23.6%	28.1%	**33.0%**
Top-20	0.2%	0.4%	1.3%	2.5%	10.8%	17.9%	27.5%	31.0%	**36.5%**
Top-50	0.5%	0.7%	2.4%	4.8%	12.2%	20.5%	29.8%	34.4%	**37.2%**
Top-100	0.8%	1.0%	4.0%	6.9%	12.8%	22.2%	31.1%	35.8%	**37.8%**
Top-200	0.9%	1.5%	5.7%	8.6%	13.6%	23.4%	32.2%	36.7%	**38.4%**
Top-500	1.3%	1.8%	6.8%	10.4%	14.4%	24.6%	33.5%	37.5%	**38.7%**

Direct GAN-based translators rank the correct category poorly because a single OBS-to-RS mapping must resolve a large and ambiguous morphological gap. Cascaded Sundial-GAN performs better than the direct GAN baselines at Top-1, while the stronger diffusion baselines substantially improve ranking quality. Among the directly comparable image-generation baselines, OBSD is strongest, reaching 69.4% validation and 20.5% held-out decipherment Top-1 accuracy. ChronoOBS reaches 86.7% and 24.4%, respectively, while also improving the higher-rank retrieval metrics. The larger improvement at Top-1 than at large k suggests that joint intermediate modeling primarily moves plausible target characters toward the front of the RS-OCR ranking. The remaining gap between Top-1 and Top-500 on held-out decipherment reflects substantial ambiguity among generated modern forms.



(a) Selected multi-period trajectories

(b) Comparison with direct generators

Figure 7: Additional qualitative results. (a) For each example, the upper row shows available reference forms across script periods, and the lower row is the path selected by ChronoOBS; black cells denote unavailable references in the upper row and inactive decoding periods in the lower row. (b) Regular Script predictions of representative generative baselines and ChronoOBS.

L Additional Qualitative Results

Figure 7(a) visualizes different intermediate-period rollouts produced by ChronoOBS. Some examples use a nearly complete modeled trajectory, whereas others activate only a subset of periods. Because path-selection frequencies alone do not establish that any particular period is causally responsible for the final prediction, we use the displayed path for visualization rather than as an attribution of unique support. Where reference glyphs are available, the visualization allows comparison of generated and authentic forms, examination of structural continuity, and localization of the stage at which a generated hypothesis begins to diverge.

Figure 7(b) illustrates different failure modes among the generative baselines. Direct GAN-based methods can collapse to similar, sparse, or structurally altered outputs, while the stronger diffusion baselines generally produce cleaner RS-like glyphs but still confuse categories requiring large morphological transformations. ChronoOBS more often preserves discriminative structure compatible with the ground-truth RS form in these examples. These qualitative cases are consistent with, but do not independently establish, the quantitative advantage reported in the main experiments.

Figure 8 exposes several recurring failure modes: sparse or highly stylized OBS strokes can be assembled into an incorrect global configuration; missing or inactive intermediate periods reduce the available structural constraints; and subtle radical or stroke distinctions can remain difficult to resolve from glyph shape alone. An incorrect final RS prediction does not necessarily imply that every generated intermediate form is uninformative: some trajectories preserve portions of the observed structure before drifting toward a competing modern form. These examples illustrate where a model-generated hypothesis becomes unreliable and reinforce the need for expert validation before any proposed decipherment is treated as established.

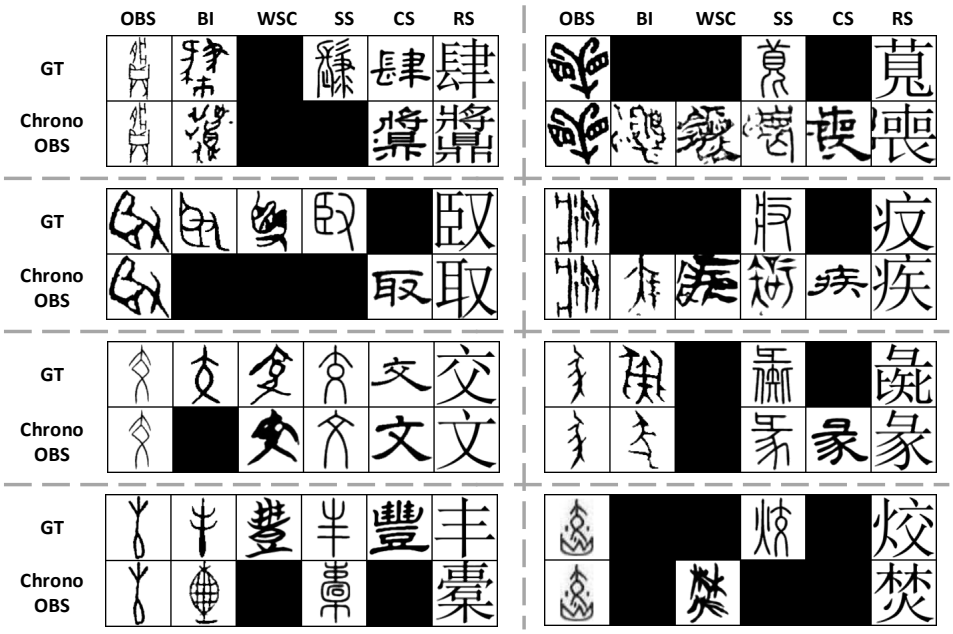


Figure 8: Representative failure cases of ChronoOBS.

M Candidate Clues for Undeciphered and Disputed OBS

Figure 9 presents selected model-generated trajectories for 36 OBS examples treated as undeciphered or disputed in the source collections used in this work. For each input, ChronoOBS generates intermediate BI, WSC, SS, and CS forms before producing an RS hypothesis. These visualizations are intended to expose the structural transformations proposed by the model and facilitate expert inspection. The generated intermediate forms are candidate reconstructions rather than verified historical attestations, and the final RS glyphs are hypotheses rather than confirmed decipherments.

Figures 10 and 11 show stochastic RS hypotheses for genuinely undeciphered OBS categories from HUST-OBC (Wang et al., 2024b) and EVOBC (Guan et al., 2024a). Because these categories have no confirmed modern labels, they are excluded from quantitative accuracy evaluation. Each row contains an authentic OBS glyph followed by ten RS hypotheses generated from different random seeds along the fixed all-period path, following Appendix I. Repeated outputs across seeds measure sampling stability rather than correctness, whereas variation reflects uncertainty in the model's generated hypotheses. These outputs are provided solely as visual candidates for paleographic assessment and require independent expert validation.



Figure 9: Predicted evolutionary trajectories for 36 undeciphered or disputed OBS characters, with two examples per row. Each example shows the input OBS glyph followed by model-generated BI, WSC, SS, CS, and RS forms (red boxes). The intermediate forms and final RS predictions are hypotheses for expert assessment, rather than established historical correspondences or confirmed decipherments.



(a) Candidate set 1 **(b)** Candidate set 2

Figure 10: Candidate clues generated for genuinely undeciphered OBS categories (Parts 1–2 of 4). Each row contains one authentic OBS glyph and ten stochastic Regular Script hypotheses.



(a) Candidate set 3 **(b)** Candidate set 4

Figure 11: Candidate clues generated for genuinely undeciphered OBS categories (Parts 3–4 of 4). Each row contains one authentic OBS glyph and ten stochastic Regular Script hypotheses.