CONTRASTIVE-TO-SELF-SUPERVISED: A TWO-STAGE FRAMEWORK FOR SCRIPT SIMILARITY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning similarity metrics for glyphs and writing systems faces a fundamental challenge: while individual graphemes within invented alphabets can be reliably labeled, the historical relationships between different scripts remain uncertain and contested. We propose a two-stage framework that addresses this epistemological constraint. First, we train an encoder with contrastive loss on labeled invented alphabets, establishing a teacher model with robust discriminative features. Second, we extend to historically attested scripts through teacher-student distillation, where the student learns unsupervised representations guided by the teacher's knowledge but free to discover latent cross-script similarities. The asymmetric setup enables the student to learn deformation-invariant embeddings while inheriting discriminative structure from clean examples. Our approach bridges supervised contrastive learning and unsupervised discovery, enabling both hard boundaries between distinct systems and soft similarities reflecting potential historical influences. Experiments on diverse writing systems demonstrate effective few-shot glyph recognition and meaningful script-level rankings without requiring ground-truth evolutionary relationships.

1 INTRODUCTION

Ancient writing systems encode millennia of human history (Kelley et al., 2025), but their visual relationships are still not fully understood (Daniels & Bright, 1996; Changizi et al., 2006; Morin, 2018; Miton & Morin, 2021). Deciphering whether two scripts share a common ancestor, or whether a glyph form migrated across cultures, is a question that archaeologists and linguists have debated for decades (Houston, 2004; Kelly et al., 2021; Desset et al., 2022; Ferrara et al., 2024). Computational approaches offer a promising route, as quantitative phylogenetics has shown for the history of languages (Gray & Atkinson, 2003; Bouckaert et al., 2012): if a model can learn a geometrically coherent similarity space over glyphs and scripts, it could provide objective, reproducible evidence to inform these debates (Rao et al., 2009; Luo et al., 2019). Yet this goal faces a fundamental conceptual challenge, one that, to our knowledge, no existing visual representation learning framework has explicitly addressed.

The obstacle is asymmetric supervision. At the glyph level, different renderings of the same character can safely be treated as positives, yielding deformation-invariant representations; and in invented alphabets, whose characters are historically independent by design, distinct classes can be treated as negatives, as many metric-learning and contrastive approaches assume (Koch et al., 2015; Snell et al., 2017; Khosla et al., 2020). Historical writing systems do not satisfy these assumptions: characters may share structural motifs, and cross-script similarities may reflect historical influence, borrowing, or common conventions, while inter-script relationships are often uncertain or only partially documented (Parpola, 1986; Duhan, 2022; Corazza et al., 2022). Asserting that two characters from different scripts are unrelated therefore risks baking in unverifiable linguistic assumptions, a domain-specific instance of the false-negative problem of contrastive learning (Chuang et al., 2020; Huynh et al., 2022).

This paper separates what can be supervised from what must remain exploratory. Our two-stage framework first learns a discriminative prior from labeled invented alphabets, whose character

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.

identities are unambiguous and historically independent, then adapts it to historical scripts without cross-script negatives. Stage 1 trains a teacher with supervised contrastive learning (Khosla et al., 2020) on fictional scripts, yielding clean intra-class clustering and inter-class separation. Stage 2 transfers this structure to unlabeled historical scripts via an asymmetric teacher-student objective inspired by BYOL (Grill et al., 2020): the student matches a momentum-updated target (Tarvainen & Valpola, 2017) under genuine handwriting variability, reorganizing where needed while preserving the teacher's geometry.

Our evaluation reflects the dual objective of the task. At the glyph level, we assess few-shot recognition via 20-way 1-shot retrieval (Lake et al., 2015; Koch et al., 2015). At the script level, we induce script-to-script distances by aggregating nearest-neighbor glyph matches and evaluate the resulting rankings against curated linguistic similarity levels using NDCG@10 (Järvelin & Kekäläinen, 2002) and Spearman correlation (Daniel, 1990). Across five backbones, our hybrid training achieves the best NDCG@10, our primary script-level metric, on three, and outperforms BYOL trained on historical scripts alone on four, while remaining competitive on glyph-level retrieval with the Simple CNN and ResNet-50 backbones. Qualitatively, the student not only inherits the teacher's discriminative structure but accentuates historically grounded proximities.

In summary, our contributions are: (i) a two-stage training strategy that decouples reliable character supervision from uncertain script relations; (ii) a teacher-initialized self-distillation adaptation that avoids cross-script negatives while allowing representations to reorganize on historical data; and (iii) an evaluation protocol combining few-shot glyph retrieval with script-level ranking against curated similarity levels, on Omniglot and a Unicode benchmark of historical scripts.

2 Related Work

Computational Paleography and Decipherment. Decipherment models align undeciphered scripts with known languages from textual evidence (Snyder et al., 2010; Luo et al., 2019; 2021), and deep networks restore and attribute ancient inscriptions (Assael et al., 2022). Visual approaches instead operate on sign shapes: glyph prototypes can be learned without supervision from document images (Siglidis et al., 2024), sign embeddings have supported the reclassification of undeciphered corpora (Corazza et al., 2022), and siamese glyph distances have been aggregated into script-level comparisons of Mediterranean alphabets (Roman & Meyer, 2024). These works analyze scripts with a given encoder; we address how to train that encoder when relations between scripts are uncertain.

Few-Shot and Metric Learning. One-shot character recognition on Omniglot (Lake et al., 2015; 2019) is typically addressed by metric learning: siamese (Koch et al., 2015; Liu et al., 2022), matching (Vinyals et al., 2016) and prototypical networks (Snell et al., 2017) learn from labeled characters and treat distinct classes as unrelated, an assumption that is safe for invented alphabets but not across historical scripts. Unsupervised meta-learning removes labels by building few-shot tasks from clustered embeddings (Hsu et al., 2019), yet its pseudo-classes still act as negatives.

Contrastive and Self-Supervised Learning. Contrastive methods such as SimCLR (Chen et al., 2020a), MoCo (He et al., 2020) and SupCon (Khosla et al., 2020) attract positive pairs while repelling all other samples as negatives, regardless of their semantic proximity (Wang & Isola, 2020). Debiased objectives (Chuang et al., 2020), hard-negative reweighting (Robinson et al., 2021) and false-negative cancellation (Huynh et al., 2022) mitigate this issue, but still assume that most negatives are unrelated. Negative-free methods avoid it altogether: Barlow Twins (Zbontar et al., 2021) reduces redundancy between views, SimSiam (Chen & He, 2021) relies on a stop-gradient, and joint-embedding predictive architectures (Assran et al., 2023; Balestriero & LeCun, 2025) predict representations in latent space.

Knowledge Distillation and Self-Distillation. Teacher–student training ranges from knowledge distillation (Hinton et al., 2015) to self-distillation (Furlanello et al., 2018), where the teacher is often a momentum (EMA) copy of the student (Tarvainen & Valpola, 2017; He et al., 2020). BYOL (Grill et al., 2020) learns invariant features by predicting an EMA teacher without negatives, and DINO/DINOv2 (Caron et al., 2021; Oquab et al., 2023) learn strong representations without annotations. Pre-training followed by teacher–student training on unlabeled data is also effective in semi-supervised learning (Xie et al., 2020; Chen et al., 2020b); in contrast, our teacher is trained on disjoint invented scripts, and its class boundaries are not transferred as hard targets.

3 METHODOLOGY

3.1 MOTIVATION FOR THE TWO-STAGE APPROACH

The fundamental challenge in studying similarities between ancient graphemes and writing systems lies in the absence of reliable ground truth regarding historical influences and evolutionary relationships. While we can confidently label individual glyphs within invented alphabets, such as the alien alphabet appearing in Matt Groening's animated series Futurama or the Tengwar script developed by J.R.R. Tolkien for the elvish languages Quenya and Telerin, we cannot make definitive claims about whether two distinct characters of historical writing systems should be considered similar or dissimilar without risking the introduction of unverifiable archaeological or linguistic assumptions. This epistemological constraint motivates our two-stage learning framework.

Stage 1: Supervised Contrastive Learning on Labeled Invented Glyphs. We first train a deep encoder (including standard CNN, Siamese, and Resnet) with contrastive loss on a curated set of invented glyphs whose ground truth labels do not introduce bias toward historical writing systems. Here, each character and its variations is considered as its own class. In addition to these handwritten variations, data augmentation (rotation, shear, zoom, translation) has been performed to preserve class identity (Koch et al., 2015). Different characters from these alphabets belong to distinct classes. This supervised phase produces a teacher model that learns robust feature representations capable of distinguishing between writing systems with known independence. Following Khosla et al. (2020), we adopt the "many-to-many" supervised contrastive loss (referred to as $\mathcal{L}^{out}$ in the original paper), which considers all instances of the same class as positives, making it particularly suited to our setting where each character class contains multiple handwritten variations and augmented samples. Glyphs from distinct classes act implicitly as negative pairs. Given a batch of B glyph images with ℓ_2-normalized embeddings $\{\mathbf{z}_i\}_{i=1}^{B}$ and labels $\{y_i\}_{i=1}^{B}$ where $y_i \in \{0, \ldots, G-1\}$ is a discrete label indicating the glyph identity of sample i, and $G = 350$ denotes the number of distinct glyph classes in the dataset of invented alphabets, the supervised contrastive loss is defined as

$$\mathcal{L}_{\text{sup}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \ell_i, \tag{1}$$

where $\mathcal{I} = \{i \mid \exists\, j \neq i \text{ such that } y_j = y_i\}$ denotes the set of anchors with at least one positive, and

$$\ell_i = -\frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp\left(\mathbf{z}_i^\top \mathbf{z}_p / \tau\right)}{\sum\limits_{a \in A(i)} \exp\left(\mathbf{z}_i^\top \mathbf{z}_a / \tau\right)}. \tag{2}$$

where $A(i) = \mathcal{I} \setminus \{i\}$, $\mathcal{P}(i) = \{p \in A(i),\ y_i = y_p\}$ and $\tau \in \mathbb{R}^*$ is a scalar temperature hyperparameter. Upon convergence, the teacher encoder f_ϕ^* yields a geometrically structured embedding space where glyph classes form well-separated clusters, as illustrated on historical scripts in Figure 3 (left). This discriminative structure serves as a semantic prior for the unsupervised adaptation stage described below. The architecture and hyperparameters of f_ϕ^* are detailed in Section 4.

Stage 2: Unsupervised Teacher-Student Distillation. However, the majority of historical writing systems cannot be cleanly separated into distinct classes due to unknown evolutionary relationships (e.g., potential influence of Phoenician on Greek) and incomplete historical records regarding script transmission. In this setting, we can only assert that augmented versions of the same character belong together, but we cannot reliably define negative pairs across historically attested scripts without introducing unverifiable linguistic assumptions. This motivates the use of a self-supervised framework that operates without explicit negative pairs (Grill et al., 2020).

BYOL Framework. Given an unlabeled dataset of glyph images $\mathcal{G}_u = \{x_i\}_{i=1}^{M}$ ($M = 922$), BYOL (Grill et al., 2020) relies on two networks sharing the same backbone architecture: an online network (student) f_θ paired with a predictor MLP q_θ, and a target network f_ξ updated via Exponential Moving Average (EMA) of the student weights:

$$\xi = \kappa \cdot \xi + (1 - \kappa) \cdot \theta, \quad \kappa \in [0, 1]. \tag{3}$$

For each image $x_i,\ i \in \{1, \ldots, M\}$ in a batch of size B', two views (x_i^1, x_i^2) are generated via data augmentation (geometric and photometric transformations). The student produces predictions

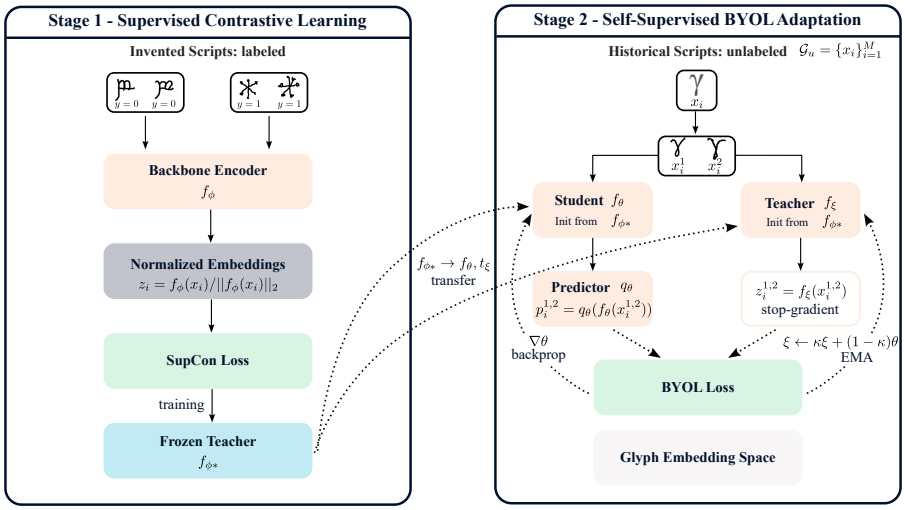


Figure 1: Two-stage framework. Stage 1: train a teacher encoder f_ϕ^* with supervised contrastive learning (SupCon) on labeled invented scripts, yielding a discriminative embedding space. Stage 2: adapt to unlabeled historical scripts via teacher-initialized BYOL: a student f_θ with predictor q_θ matches a momentum (EMA) target network f_ξ on two augmented views, using stop-gradient, and without cross-script negatives.

through the predictor: $p_i^1 = q_\theta(f_\theta(x_i^1))$ and $p_i^2 = q_\theta(f_\theta(x_i^2))$ while the target network produces representations with a stop-gradient (sg): $z_i^1 = \text{sg}(f_\xi(x_i^1))$ and $z_i^2 = \text{sg}(f_\xi(x_i^2))$. The training objective minimizes the symmetrized negative cosine similarity between student predictions and target representations:

$$\mathcal{L}_{\text{BYOL}} = \frac{1}{B'} \sum_{i=1}^{B'} \left[\mathcal{D}(p_i^1, z_i^2) + \mathcal{D}(p_i^2, z_i^1) \right] \tag{4}$$

where $\mathcal{D}(p, z) = 2 - 2 \cdot \frac{p^\top z}{\|p\|_2 \cdot \|z\|_2}$. The asymmetry between the student (with predictor) and the target network (without predictor, no gradient flow) prevents representational collapse in the embedding space without requiring negative pairs.

Our adaptation. Our approach departs from standard BYOL in three key ways, illustrated in Figure 1. First, rather than initializing both networks randomly, the student f_θ and the target network f_ξ are both initialized from the teacher f_ϕ^* obtained at the end of Stage 1. This provides a semantically structured initialization that guides the student toward meaningful representations from the very first iterations, as opposed to standard BYOL which must discover representational structure from scratch without any semantic prior.

Second, regarding the network architecture, we deliberately omit the projection MLP typically inserted between the backbone and the predictor in the original BYOL formulation. Since our backbone already produces compact low-dimensional embeddings ($d \in \{64, 128, 256\}$, see Section 4.1), an additional projector would add unnecessary complexity without meaningful benefit, and would risk overfitting given the moderate size of the Omniglot dataset. The predictor q_θ thus operates directly on the backbone embeddings.

Third, rather than generating two augmented views from a single image as in standard BYOL, we leverage the availability of multiple genuine handwritten instances per character class, complemented by the same geometric augmentations (rotation, shear, zoom, translation) applied in Stage 1.

3.2 Glyph and Script Similarity

Glyph-Level Similarity. Once the model has been trained, it defines an embedding space where each glyph x_i is mapped to a d-dimensional feature vector z_i and ℓ_2-normalized. For any pair of glyph images (x_1, x_2), we compute their similarity as the cosine of their embeddings, and their

dissimilarity as follows:

$$\begin{aligned} \mathrm{sim}(x_1, x_2) &= \cos\big(z_1, z_2\big) = z_1^\top z_2, \\ d_g(x_1, x_2) &:= 1 - \mathrm{sim}(x_1, x_2). \end{aligned} \tag{5}$$

This quantity reflects the angular proximity between glyph representations in the learned feature space. This procedure is applied independently for each trained model, thereby inducing a model-specific geometry on the space of glyphs.

Script-Level Similarity. A script s is represented as a finite set of glyphs. To compare two scripts s_1 and s_2, we aggregate glyph-level dissimilarities through a nearest-neighbor matching scheme. First, we define the directed average distance

$$\tilde{d}_s(s_1, s_2) := \frac{1}{|s_1|} \sum_{x_1 \in s_1} \min_{x_2 \in s_2} d_g(x_1, x_2), \tag{6}$$

which measures how well each glyph of s_1 can be approximated by its closest counterpart in s_2. To obtain a symmetric script-level dissimilarity, we define

$$d_s(s_1, s_2) := \frac{1}{2}\Big(\tilde{d}_s(s_1, s_2) + \tilde{d}_s(s_2, s_1)\Big). \tag{7}$$

This formulation allows multiple glyphs from one script to be associated with the same glyph in another to reflect the historical evolution of writing systems (Roman & Meyer, 2024).

3.3 Evaluation Metrics

20-Way 1-Shot Evaluation. At the glyph level, representation quality is evaluated with a 20-way 1-shot retrieval task (Lake et al., 2015; Koch et al., 2015). For each episode, $N = 20$ classes are sampled; a query glyph is matched against one positive candidate (same class) and $N - 1$ negatives (one per other class), ranked by the similarity of Equation (5). We report Top-k accuracy ($k \in \{1, 5\}$) averaged over 400 episodes.

Normalized Discounted Cumulative Gain. Beyond character-level discrimination, we assess whether the representations capture relationships between writing systems. Given a query script s_q, scripts are ranked by increasing $d_s(s_q, \cdot)$, and ground-truth relationships are encoded as discrete similarity levels (lower indicates stronger similarity) converted into graded relevance scores. The Discounted Cumulative Gain (Järvelin & Kekäläinen, 2002) at rank k and its normalized form are defined as

$$\begin{aligned} \mathrm{DCG@}k &= \sum_{r=1}^{k} \frac{\mathrm{rel}_r}{\log_2(r+1)}, \\ \mathrm{NDCG@}k &= \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k}, \end{aligned} \tag{8}$$

where rel_r denotes the relevance of the script ranked at position r. Normalization by the Ideal DCG (IDCG), obtained from the ground-truth ordering, ensures comparability across queries. We report NDCG@10 averaged over all query scripts.

Spearman's Rank Correlation Coefficient. To quantify the global agreement between embedding-induced script geometry and ground-truth levels, we compute Spearman's rank correlation ρ (Daniel, 1990) between the predicted dissimilarities $d_s(s_i, s_j)$ and the similarity levels $\ell(s_i, s_j)$ over all evaluated script pairs. Since smaller levels correspond to stronger relationships, a positive and statistically significant ρ indicates that historically closer scripts are embedded closer together.

3.4 Datasets

Omniglot. The Omniglot dataset (Lake et al., 2015) consists of 1,623 handwritten character classes drawn from 50 different writing systems from around the world (details are given in Appendix A). Each character class consists of 20 instances produced by different human subjects and each glyph is

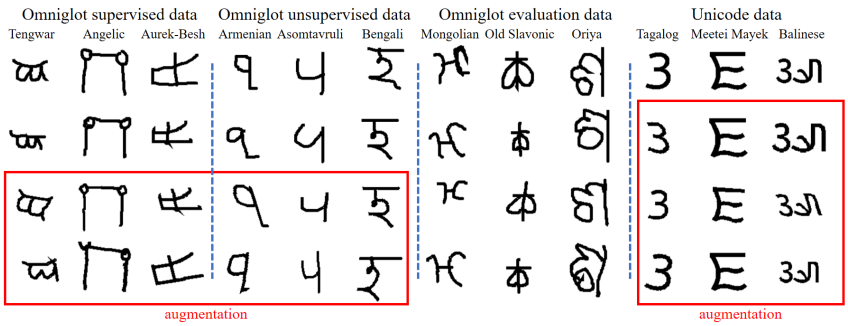


Figure 2: Datasets used in this work. Omniglot is split into supervised (invented alphabets for Stage 1), unsupervised (historical scripts for Stage 2), and evaluation sets. A complementary Unicode character dataset is generated using Noto fonts. Augmented instances per glyph are obtained through random transformations (red boxes).

represented as a black-and-white 105×105 pixels image. To augment the dataset we generated 8 additional instances per glyph via random affine perturbations (Koch et al., 2015) (rotation, shear, zoom, and small translations), each applied independently with probability 0.5. The dataset contains historically attested scripts and fictional or modern alphabets introduced after the 18th century. We separate it in three distinct datasets (see Figure 2):

- the Omniglot supervised invented dataset composed of the 15 fictional or modern scripts, dedicated for supervised learning,
- the Omniglot unsupervised dataset composed of 25 historically attested scripts, dedicated for unsupervised learning,
- the Omniglot evaluation dataset composed of 10 historically attested scripts.

Unicode. To further evaluate the models we construct a glyph and script dataset by considering all writing systems whose historical development predates the 19th century from the Unicode 17.0 repertoire (The Unicode Consortium, 2025). For each script, its character inventory is defined using explicit Unicode codepoint ranges and graphemes are rendered using the Google Noto font family (Google Inc., 2012). This method offers broad and consistent coverage across scripts while limiting stylistic variation. Each Unicode character is centered using its non-white bounding box, with font size adaptively adjusted to prevent clipping and preserve comparable stroke widths, and rendered as a black-and-white 105×105 pixels image, see Figure 2.

Ground-truth relationships between scripts are encoded as three discrete similarity levels derived from historical and typological knowledge of writing system evolution (Daniels & Bright, 1996). Level 1 (very similar) corresponds to scripts that are essentially variants of the same writing system or closely related historical forms, for example Georgian Asomtavruli and Georgian Mkhedruli, or Hiragana and Katakana. Level 2 (similar) groups scripts belonging to the same historical lineage or direct derivational branch, such as Greek and Cyrillic, or Brahmi and Devanagari. Level 3 (very different) corresponds to scripts from unrelated families with no clear historical derivation, for instance Latin and Hangul, or Devanagari and Chinese characters. Pairs not explicitly assigned to Levels 1–3 are treated as unrelated scripts and receive the lowest similarity level during evaluation to avoid introducing speculative historical relationships and permit to evaluate on the full set of script pairs. The complete list of Unicode scripts and their assigned similarity levels is provided in Appendix A.3.

4 Experimental Results

4.1 Experimental Setup

Baseline SotA. We compare our hybrid framework against two self-supervised baselines trained without labels: BYOL (Grill et al., 2020) (Section 3.1) and Barlow Twins (Zbontar et al., 2021), which minimizes feature redundancy between two augmented views. Both test whether supervised

pre-training helps over purely self-supervised training. We further include DINOv2-ViT-S/14 (Oquab et al., 2023), a foundation model trained on 142M images, evaluated frozen and with a BYOL-fine-tuned head, to test whether generic features suffice for ancient script analysis.

To isolate Stage 2, we also evaluate the frozen teacher f_ϕ^* (end of Stage 1) without BYOL adaptation. Since Stage 2 yields two networks, the gradient-trained student f_θ and the EMA target f_ξ (known to produce smoother representations (Grill et al., 2020)), we report both at inference.

Data Splits. Following the dataset description in Section 3.4, Stage 1 uses the Omniglot supervised dataset (15 fictional and modern scripts, $G = 350$ character classes) with labeled instances. For the hybrid approach, Stage 2 uses the Omniglot unsupervised dataset (25 historically attested scripts) as unlabeled data. For the baselines (BYOL simple, Barlow Twins), both the supervised and unsupervised datasets are combined into a single unlabeled pool, ensuring a fair comparison, our hybrid approach has access to the same data overall. Glyph-level retrieval is evaluated on the Omniglot evaluation dataset (10 historically attested scripts, unseen during training), and script-level metrics (NDCG@10, Spearman) on the Unicode dataset. For Stage 1, 10% of characters are held out for validation at the character level to avoid class overlap.

Backbones. We evaluate five backbone architectures of increasing capacity: a Simple CNN (0.2M parameters), a Siamese-like CNN (2.2M) (a branch of the network of Koch et al. (2015)), ResNet-18 (11.2M), ResNet-34 (21.3M), and ResNet-50 (23.8M) (He et al., 2016). Embeddings have dimension $d \in \{64, 128, 256\}$, except for the Siamese CNN ($d = 4096$, sigmoid output, following Koch et al. (2015)), and are ℓ_2-normalized before similarity computation. We additionally evaluate DINOv2-ViT-S/14 (Oquab et al., 2023) (22M parameters) as a large pretrained baseline, both frozen and fine-tuned. Note that BYOL (Grill et al., 2020) and Barlow Twins (Zbontar et al., 2021) were originally proposed with a ResNet-50 backbone; we therefore include ResNet-50 as a common reference point for these baselines alongside our hybrid approach. More details about the architectures are provided in Appendix B.1.

Hyperparameter Optimization. All methods are tuned with Optuna (Akiba et al., 2019) (TPE sampler, median pruner, 30 trials) by maximizing 1-NN validation accuracy, over batch size $\{128, 256, 512\}$, learning rate $[3 \times 10^{-5}, 3 \times 10^{-4}]$, weight decay $[10^{-7}, 10^{-4}]$, plus method-specific terms (temperature, EMA decay, $\lambda_{\text{off-diag}}$). Optimal configurations consistently favored learning rates around 10^{-4}, small weight decay, and long schedules (300–500 epochs). The full search space and best configuration for every study are reported in Appendix B.2.

Implementation Details. All models are trained with the AdamW optimizer using a warmup and cosine decay learning rate schedule. At each training step, a single pair (x_1, x_2) of distinct handwritten instances is sampled per character class, leveraging the multiple genuine handwritten variations available in Omniglot rather than relying solely on synthetic augmentations.

4.2 Benchmark

Primary Metric: Script-Level Ranking Quality. Our primary objective is to recover an embedding space that reflects the historical and linguistic relationships between writing systems, so we treat NDCG@10 as our primary metric, as it rewards ranking historically related scripts among a query's nearest neighbours, and Spearman's ρ as a complementary metric measuring the global monotonic agreement between induced dissimilarities and ground-truth levels.

As shown in Table 1, our hybrid approach achieves the highest NDCG@10 on three backbones (Simple CNN, ResNet-34, ResNet-50), the remaining cases going to SupCon (Siamese CNN) and Barlow Twins (ResNet-18). This confirms that the Stage-1 semantic prior aligns the embedding space with historical relationships better than purely self-supervised methods, with the highest NDCG@10 of the whole table on ResNet-50 (0.3178 vs. 0.2997 for Barlow Twins and 0.2708 for BYOL).

On Spearman, our method is best on Simple CNN ($\rho = 0.640$) but trails the baselines on the other backbones, and is lowest on the three ResNets. This is consistent with Spearman measuring global rank agreement across all script pairs, whereas NDCG@10 targets the top of the ranking, which our hybrid initialization improves.

Secondary Metrics: Glyph-Level Retrieval. At the glyph level, Top-1/Top-5 measure 20-way 1-shot retrieval accuracy. Although not our primary objective, they gauge the discriminative quality of the representations. On Simple CNN and ResNet-50 our approach is competitive with or better than all baselines (e.g., Simple CNN: 88.0/98.75, above SupCon, BYOL, and Barlow Twins), showing it does not trade glyph-level discrimination for script-level coherence.

On ResNet-18 and ResNet-34, Barlow Twins and BYOL reach higher Top-1: on these mid-size ResNets the SupCon teacher itself is weaker (64 and 73 Top-1), and Stage 2 recovers part of the gap on ResNet-34 (84.25) but not on ResNet-18. Yet our method keeps the best NDCG@10 on ResNet-34, so the script-level geometry can remain coherent even when glyph-level Top-1 is not maximised.

Finally, DINOv2-ViT-S/14, despite its 22M parameters, achieves modest glyph-level retrieval and NDCG@10 in both frozen and fine-tuned configurations. This confirms that large vision transformers pre-trained on natural images do not transfer well to the highly specific domain of ancient script analysis, and underlines the importance of domain-adapted training strategies such as the one proposed in this work.

This trade-off is further reinforced by two additional experiments reported in Appendix C.1: a BYOL variant trained only on Stage-2 data, without the Stage-1 prior, and a fully fine-tuned DINOv2 both reach higher glyph-level accuracy than their counterparts in Table 1 yet obtain a lower NDCG@10 than our method on most backbones, confirming that high glyph discrimination does not translate into a coherent script-level geometry.

Table 1: Comparison of backbone architectures and training strategies on glyph- and script-level evaluation. We report 20-way 1-shot retrieval accuracy (N20R1/N20R5), script-level ranking quality (NDCG@10), and Spearman correlation with curated linguistic similarity levels. Retrieval is evaluated on the Omniglot evaluation set, script-level metrics on the Unicode dataset. "Teacher Type" indicates whether an EMA teacher is used; here, the teacher is obtained by supervised contrastive pretraining followed by EMA self-distillation (SupCon→EMA). SupCon denotes the frozen Stage-1 teacher f_ϕ^*. Best results are in **bold**, second best are underlined.

Backbone	Params	Method	Teacher Type	N20R1	N20R5	NDCG10	Spearman
Simple CNN	0.2M	SupCon	-	85.5	**98.75**	0.2781	0.548
		BYOL	EMA	43.5	83.75	0.1932	0.545
		Barlow Twins	-	77.75	97.75	0.2281	0.639
		Ours (Target)	SupCon→EMA	**88**	**98.75**	**0.2930**	**0.640**
		Ours (Student)	SupCon→EMA	**88**	**98.75**	**0.2930**	0.639
Siamese CNN	2.2M	SupCon	-	86.75	99.5	**0.3011**	0.631
		BYOL	EMA	75.00	94.75	0.2716	**0.642**
		Barlow Twins	-	**94.25**	**99.75**	0.2665	0.536
		Ours (Target)	SupCon→EMA	85	99	0.2677	0.594
		Ours (Student)	SupCon→EMA	85	99	0.2677	0.594
ResNet-18	11.2M	SupCon	-	64	84.75	0.2480	0.471
		BYOL	EMA	89.25	99.50	0.2882	**0.617**
		Barlow Twins	-	**93.75**	**99.75**	**0.3031**	0.431
		Ours (Target)	SupCon→EMA	63.25	86.75	0.2537	0.379
		Ours (Student)	SupCon→EMA	63.5	85	0.2532	0.373
ResNet-34	21.3M	SupCon	-	73	94.5	0.2647	0.433
		BYOL	EMA	86.25	98.00	0.2428	0.445
		Barlow Twins	-	**93.00**	**99.5**	0.2895	**0.482**
		Ours (Target)	SupCon→EMA	70.75	92.75	0.2518	0.383
		Ours (Student)	SupCon→EMA	84.25	96.75	**0.2927**	0.388
ResNet-50	23.8M	SupCon	-	86.25	97.50	0.3078	0.514
		BYOL	EMA	87.50	99.25	0.2708	**0.622**
		Barlow Twins	-	**93.75**	**100**	0.2997	0.481
		Ours (Target)	SupCon→EMA	93.00	98.75	**0.3178**	0.424
		Ours (Student)	SupCon→EMA	93.00	98.75	**0.3178**	0.424
DINOv2-ViT-S 14	22M	Frozen	-	43.75	74.75	0.2612	0.477
		Fine-tuned	-	61	90.5	0.2366	0.609

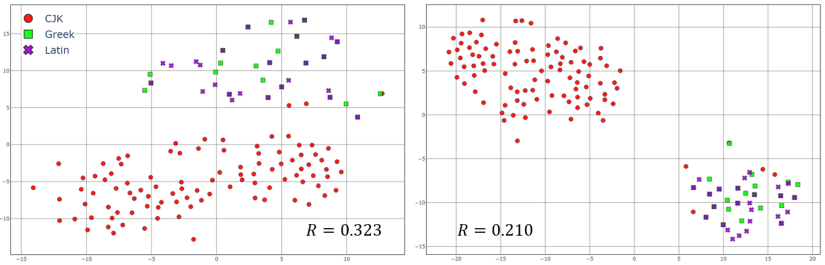


Figure 3: Two-dimensional t-SNE projections (Van der Maaten & Hinton, 2008) of glyph embeddings for the CJK, Greek and Latin scripts, shown on the left using the teacher model f_ϕ^* and on the right the student model f_θ. Points correspond to glyph instances; colors/markers indicate script labels. The reported $\mathcal{R}$ denotes the separability ratio (9), illustrating how improved representation geometry yields more coherent cross-script structure.

Separability Ratio. To further assess the geometric quality of the learned embedding space, we introduce a separability ratio $\mathcal{R}$ that measures how much closer linguistically related scripts are compared to unrelated ones. Given three scripts, for instance Greek, Latin, and CJK (Chinese-Japanese-Korean) characters, where Greek and Latin share a common historical origin while CJK is linguistically independent, we define

$$\mathcal{R} = \frac{d_s(\text{Greek}, \text{Latin})}{\frac{1}{2}\left[d_s(\text{CJK}, \text{Greek}) + d_s(\text{CJK}, \text{Latin})\right]} \tag{9}$$

A lower value of $\mathcal{R}$ indicates that linguistically related scripts (Greek, Latin) are embedded proportionally closer to each other than to an unrelated script (CJK), reflecting a geometrically coherent organisation of the embedding space with respect to historical relationships. As shown in Figure 3, the teacher f_ϕ^* yields $\mathcal{R} = 0.323$, while the student f_θ achieves $\mathcal{R} = 0.210$, a relative reduction of 35%. This result demonstrates that the unsupervised adaptation in Stage 2 does not merely compress the embedding space uniformly, but selectively accentuates historically grounded proximities, producing a geometry that better reflects the linguistic structure of writing systems.

5 Conclusion

We studied glyph and script similarity under asymmetric supervision: character identity can be safely supervised in invented alphabets, whereas cross-script relations in historical writing systems are uncertain and should not be enforced through negative pairs. We proposed a two-stage framework that learns a discriminative prior via supervised contrastive training on invented scripts, then adapts to historical scripts through teacher-initialized self-distillation without cross-script negatives.

Across five backbones and two datasets (Omniglot and our Unicode benchmark), this strategy improves script-level ranking, achieving the best NDCG@10 on three of the five backbones, including the highest overall NDCG@10 on ResNet-50, while remaining competitive on 20-way 1-shot retrieval with the Simple CNN and ResNet-50 backbones. The separability ratio and t-SNE analyses indicate that Stage 2 sharpens historically grounded proximities rather than merely compressing the teacher space, and comparisons with BYOL, Barlow Twins, SupCon, and DINOv2 show that domain-adapted training is crucial beyond generic visual features.

Our approach has limitations. The Stage 1 prior depends on how representative invented alphabets are of historical scripts, and on ResNet-18 and ResNet-34 the weaker SupCon teacher limits glyph-level Top-1, while Spearman correlation remains below the baselines on ResNet backbones. The three-level ground truth simplifies contested relationships, evaluation is restricted to Omniglot and a single-font Unicode benchmark, and each configuration is a single run without multi-seed error bars. Future work will evaluate on real manuscripts and degraded inscriptions, report multi-seed variance, use the induced script distance for broader phylogenetic analyses of writing systems, and enrich training with learned augmentations (e.g., generative synthesis of handwriting variation). More broadly, the same principle, reliable within-class supervision then adaptation without speculative negatives, could transfer to settings with uncertain cross-category relations.

ETHICS STATEMENT

Our study analyses images of glyphs from publicly available resources and does not involve human subjects or personally identifiable information. We use the underlying artifacts in accordance with their licenses: Omniglot is released under the MIT license and the Google Noto fonts under the SIL Open Font License, both of which permit research use and the redistribution of derived glyph images. The principal risk we foresee is interpretive: the script-level proximities produced by our models are statistical regularities in a learned visual space and should not be read as definitive evidence of historical filiation or borrowing between writing systems. We therefore present them as a tool to inform, not replace, expert epigraphic and linguistic analysis. Finally, training relied on modest compute: all experiments ran on a single cloud GPU (Google Colab), with each per-model Optuna study (30 trials) taking approximately 5–8 hours, limiting the environmental footprint of this work.

REPRODUCIBILITY STATEMENT

All the elements needed to reproduce our results are described in the paper and its appendices. The three Omniglot splits and the construction of the Unicode benchmark, including the complete script inventories, are detailed in Section 3.4 and Appendix A; the ground-truth similarity levels and the full list of annotated script pairs are given in Appendix A.3; the backbone architectures and the embedding normalization are specified in Appendix B.1; and the Optuna search spaces together with the selected hyperparameters of every study are reported in Appendix B.2. The training protocol of both stages, including losses, augmentations and data splits, is given in Section 3.1 and Section 4.1. The source code used to build the datasets, train the models and compute all reported metrics is provided as anonymized supplementary material.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the writing of the manuscript (grammar, phrasing and formatting) and as a coding aid when implementing training and evaluation scripts. We did not use generative AI tools for research ideation or for producing experimental results: the two-stage framework, the evaluation protocol, and all reported numbers stem from experiments designed and run by the authors. The curation of the ground-truth script similarity levels was carried out manually by the authors on the basis of the linguistic literature. We have reviewed all AI-assisted work: AI-assisted text was checked and rewritten by the authors, and AI-assisted code was verified and tested against the reported results. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Takeru Akiba, Shotaro Sano, Toshiki Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *ACM SIGKDD*, 2019.

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(7900):280–283, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, June 2023.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Remco Bouckaert, Philippe Lemey, Michael Dunn, Simon J Greenhill, Alexander V Alekseyenko, Alexei J Drummond, Russell D Gray, Marc A Suchard, and Quentin D Atkinson. Mapping the origins and expansion of the Indo-European language family. *Science*, 337(6097):957–960, 2012. doi: 10.1126/science.1219669. URL https://www.science.org/doi/abs/10.1126/science.1219669.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

Mark A Changizi, Qiong Zhang, Hao Ye, and Shinsuke Shimojo. The structures of letters and symbols throughout human history are selected to match those found in objects in natural scenes. *The American Naturalist*, 167(5):E117–E139, 2006. doi: 10.1086/502806. URL https://www.journals.uchicago.edu/doi/abs/10.1086/502806.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22243–22255. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fcbc95ccdd551da181207c0c1400c655-Paper.pdf.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, June 2021.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Michele Corazza, Fabio Tamburini, Miguel Valério, and Silvia Ferrara. Unsupervised deep learning supports reclassification of Bronze Age Cypriot writing system. *PloS one*, 17(7):e0269544, 2022.

W.W. Daniel. *Applied Nonparametric Statistics*. Duxbury advanced series in statistics and decision sciences. PWS-KENT Pub., 1990. ISBN 9780534919764. URL https://books.google.fr/books?id=0hPvAAAAMAAJ.

Peter T. Daniels and William Bright (eds.). *The World's Writing Systems*. Oxford University Press, New York, 1996. ISBN 978-0-19-507993-7.

François Desset, Kambiz Tabibzadeh, Matthieu Kervran, Gian Pietro Basello, and Gianni Marchesi. The decipherment of Linear Elamite writing. *Zeitschrift für Assyriologie und vorderasiatische Archäologie*, 112(1):11–60, 2022.

Paras Duhan. Origin of Brahmi script from logographic elements: an analysis. *Integr J Res Arts Humanit*, 2(5):18–24, 2022.

Silvia Ferrara, Laura Tassoni, Bernd Kromer, Lukas Wacker, Michael Friedrich, Francesca Tonini, Lorenzo Lastilla, Roberta Ravanelli, and Sahra Talamo. The invention of writing on Rapa Nui (Easter Island). New radiocarbon dates on the Rongorongo script. *Scientific Reports*, 14(1):2794, Feb 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-53063-7. URL https://doi.org/10.1038/s41598-024-53063-7.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

Google Inc. Noto fonts. https://www.google.com/get/noto/, 2012. Accessed: 2026-01.

Russell D. Gray and Quentin D. Atkinson. Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, 426(6965):435–439, Nov 2003. ISSN 1476-4687. doi: 10.1038/nature02029. URL https://doi.org/10.1038/nature02029.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Stephen D Houston. *The first writing: Script invention as history and process*. Cambridge University Press, 2004.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*, 2019.

Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 986–996, 2022. doi: 10.1109/WACV51458.2022.00106.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Kathryn Kelley, Mattia Cartolano, and Silvia Ferrara. Seals and signs: tracing the origins of writing in ancient South-west Asia. *Antiquity*, 99(403):64–82, 2025.

Piers Kelly, James Winters, Helena Miton, and Olivier Morin. The predictable evolution of letter shapes: An emergent script of West Africa recapitulates historical change in writing systems. *Current Anthropology*, 62(6):669–691, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pp. 1–30. Lille, 2015.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The Omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2019.04.007. Artificial Intelligence.

Xuxing Liu, Weize Gao, Rankang Li, Yu Xiong, Xiaoqin Tang, and Shanxiong Chen. One shot ancient character recognition with siamese similarity network. *Scientific reports*, 12(1):14820, 2022.

Jiaming Luo, Yuan Cao, and Regina Barzilay. Neural decipherment via minimum-cost flow: From Ugaritic to Linear B. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3146–3155, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1303. URL https://aclanthology.org/P19-1303/.

Jiaming Luo, Frederik Hartmann, Enrico Santus, Regina Barzilay, and Yuan Cao. Deciphering undersegmented ancient scripts using phonetic prior. *Transactions of the Association for Computational Linguistics*, 9:69–81, 2021. doi: 10.1162/tacl_a_00354. URL https://aclanthology.org/2021.tacl-1.5/.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.

Helena Miton and Olivier Morin. Graphic complexity in writing systems. *Cognition*, 214:104771, 2021.

Olivier Morin. Spontaneous emergence of legibility in writing systems: The case of orientation anisotropy. *Cognitive Science*, 42(2):664–677, 2018.

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. *arXiv preprint arXiv:2304.07193*, 2023.

Asko Parpola. The Indus script: A challenging puzzle. *World Archaeology*, 17(3):399–419, 1986. doi: 10.1080/00438243.1986.9979979. URL https://doi.org/10.1080/00438243.1986.9979979.

Rajesh PN Rao, Nisha Yadav, Mayank N Vahia, Hrishikesh Joglekar, Ronojoy Adhikari, and Iravatham Mahadevan. Entropic evidence for linguistic structure in the Indus script. *Science*, 324 (5931):1165–1165, 2009.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CR1XOQ0UTh-.

Claire Roman and Philippe Meyer. Analysis of glyph and writing system similarities using Siamese neural networks. In Rachele Sprugnoli and Marco Passarotti (eds.), *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, pp. 98–104, Torino, Italia, May 2024. ELRA and ICCL.

Ioannis Siglidis, Nicolas Gonthier, Julien Gaubil, Tom Monnier, and Mathieu Aubry. The learnable typewriter: a generative approach to text analysis. In *Document Analysis and Recognition - ICDAR 2024: 18th International Conference, Athens, Greece, August 30–September 4, 2024, Proceedings, Part II*, pp. 297–314, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-70535-9. doi: 10.1007/978-3-031-70536-6_18. URL https://doi.org/10.1007/978-3-031-70536-6_18.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Benjamin Snyder, Regina Barzilay, and Kevin Knight. A statistical model for lost language decipherment. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre (eds.), *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1048–1057, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1107/.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/68053af2923e00204c3ca7c6a3150cf7-Paper.pdf.

The Unicode Consortium. The Unicode standard, version 17.0. https://www.unicode.org/versions/Unicode17.0.0/, 2025. Accessed: 2026-01.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves ImageNet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

A Datasets

A.1 Omniglot Splits

The training and the evaluation of the models considered in the paper rely on disjoint collections of scripts from the Omniglot dataset (Lake et al., 2015). Each script corresponds to an alphabet or writing system containing multiple characters. Each character is associated with several rendered instances used for training or evaluation. The scripts are divided as follows:

- Teacher training set used for supervised contrastive learning in Stage 1 (Table 2).
- Student training set used for self-supervised learning in Stage 2 (Table 3).
- Evaluation set used exclusively for glyph-level retrieval (Table 4).

Table 2: Scripts used for teacher training.

Script name	**Number of glyphs**
Alphabet of the Magi	20
Angelic	20
Arcadian	26
Atemayar Qelisayer	26
Atlantean	26
Aurek-Besh	26
Blackfoot (Canadian Aboriginal Syllabics)	14
Braille	26
Futurama	26
Inuktitut (Canadian Aboriginal Syllabics)	16
Keble	26
N'Ko	33
Ojibwe (Canadian Aboriginal Syllabics)	14
Tengwar	25
ULOG	26

Table 3: Scripts used for student training.

Script name	**Number of glyphs**
Anglo-Saxon Futhorc	29
Armenian	41
Asomtavruli (Georgian)	40

Avesta	26
Balinese	24
Bengali	46
Burmese (Myanmar)	34
Cyrillic	33
Early Aramaic	22
Ge‘ez	26
Glagolitic	45
Grantha	43
Greek	24
Gujarati	48
Gurmukhi	45
Hebrew	22
Japanese (Hiragana)	52
Japanese (Katakana)	47
Kannada	41
Korean	40
Latin	26
Malay (Jawi - Arabic)	40
Malayalam	47
Manipuri	40
Mkhedruli (Georgian)	41

Table 4: Scripts used for evaluation.

Script name	**Number of glyphs**
Mongolian	30
Old Church Slavonic (Cyrillic)	45
Oriya	46
Sanskrit	42
Sylheti	28
Syriac (Estrangelo)	23
Syriac (Serto)	23
Tagalog	17
Tibetan	42
Tifinagh	55

A.2 UNICODE BENCHMARK

Table 5 summarizes the metadata for all scripts included in the Unicode dataset. For each script, we report the number of glyph classes retained, large scripts are capped at $k = 100$ representative glyphs.

Table 5: Size metadata for all scripts included in the Unicode dataset. For each script, we report the number of glyph classes retained. An asterisk (*) indicates scripts for which the glyph inventory was limited to 100 representative glyphs.

Script name	**Number of glyphs**
Ahom	27
Anatolian Hieroglyphs	100*
Arabic	36
Armenian	38
Avestan	54
Balinese	47
Batak	38
Bengali	44

Bhaiksuki	46
Brahmi	51
Buginese	23
Buhid	18
CJK	100*
Carian	49
Caucasian Albanian	52
Chakma	36
Cham	41
Chorasmian	21
Coptic	25
Cuneiform	100*
Cypriot Syllabary	55
Cypro-Minoan	97
Cyrillic	32
Devanagari	54
Dogra	44
Egyptian Hieroglyphs	100*
Elder Futhark	24
Elymaic	22
Ge‘ez	26
Georgian Asomtavruli	38
Georgian Mkhedruli	33
Glagolitic	47
Gothic	27
Grantha	46
Greek	24
Gujarati	48
Gunjala Gondi	40
Gurmukhi	43
Hangul Jamo	67
Hanunoo	18
Hatran Aramaic	21
Hebrew	27
Hiragana	74
Imperial Aramaic	22
Javanese	47
Kaithi	45
Kanbun	16
Kannada	49
Katakana	78
Kawi	47
Kharoshthi	37
Khmer	50
Khojki	43
Khudawadi	47
Lao	41
Latin	26
Lepcha	36
Limbu	31
Linear A	100*
Linear B	100*
Lycian	29
Lydian	26
Mahajani	35
Makasar	19
Malayalam	52
Mandaic	25
Manichaean	37

Marchen	30
Meetei Mayek	35
Meroitic Cursive	24
Meroitic Hieroglyphs	32
Modi	48
Mongolian	35
Multani	37
Myanmar	43
Nabataean	31
Nandinagari	47
Newa	53
Ogham	26
Old Hungarian	51
Old Italic	27
Old North Arabian	29
Old Permic	38
Old Persian	36
Old Sogdian	18
Old South Arabian	29
Old Turkic Orkhon	42
Old Turkic Yenisei	31
Old Uyghur	18
Oriya	46
Pahlavi Inscriptional	19
Pahlavi Psalter	18
Palmyrene	23
Parthian Inscriptional	22
Phags-pa	39
Phaistos Disc	45
Phoenician	22
Rejang	23
Samaritan	22
Sharada	48
Siddham	47
Sinhala	59
Sogdian	22
Soyombo	41
Sundanese	30
Syloti Nagri	27
Syriac	26
Tagalog	18
Tagbanwa	16
Tai Le	30
Tai Tham	53
Tai Viet	48
Takri	43
Tamil	35
Tangut	100*
Telugu	50
Thaana	24
Thai	47
Tibetan	42
Tifinagh	56
Tirhuta	47
Todhri	52
Ugaritic	30
Yezidi	42
Zanabazar Square	41

A.3 Ground-Truth Similarity Levels

This subsection provides the complete list of ground-truth script similarity pairs used for evaluation of script-level distances. As described in Section 3.4, script relationships are encoded using three discrete similarity levels derived from historical and typological knowledge of writing system evolution (Daniels & Bright, 1996).

The objective of this hierarchy is not to reconstruct a precise phylogenetic tree of writing systems, which remains debated in historical linguistics, but rather to provide a coarse, interpretable ordering of script proximity suitable for evaluating whether the learned embedding space captures historically plausible relationships.

We define three similarity levels:

- **Level 1 (very similar).** Scripts that correspond to regional, stylistic, or temporal variants of the same writing system. These scripts typically share the same structural principles and many glyph shapes. Examples include *Hiragana–Katakana* or *Georgian Asomtavruli–Georgian Mkhedruli* (see Table 6).
- **Level 2 (similar).** Scripts belonging to the same historical lineage or direct derivational branch. These relationships correspond to well-documented genealogical links in the history of writing systems, such as *Greek–Cyrillic* or *Brahmi–Devanagari* (see Table 7).
- **Level 3 (very different).** Scripts belonging to independent families with no known direct derivation or shared structural origin. Examples include *Latin–Hangul* or *Devanagari–CJK* (see Table 8).

Pairs not explicitly assigned to one of these levels are treated as unrelated scripts during evaluation and assigned the lowest similarity level, ensuring that the model is evaluated across the full space of script combinations without introducing speculative historical assumptions.

Construction of the Pair List. The pair list was manually curated using historical and typological references on the evolution of writing systems, including standard works on script genealogy and classification. The construction follows three principles:

1. **Conservative historical relations.** Only well-established relationships documented in the literature were included in Levels 1 and 2.
2. **Family-level diversity.** Level 3 pairs were selected to cover scripts from independent writing traditions, including Phoenician-derived alphabets, Brahmic scripts, East Asian logographic systems, Korean Hangul, and several ancient or undeciphered scripts.
3. **Coverage of Unicode scripts.** Pairs were constructed to ensure broad coverage of the scripts included in the Unicode evaluation dataset.

Table 6: Level 1: very similar script pairs.

Script A	Script B
Georgian Asomtavruli	Georgian Mkhedruli
Cyrillic	Glagolitic
Imperial Aramaic	Hatran Aramaic
Imperial Aramaic	Palmyrene
Hatran Aramaic	Palmyrene
Pahlavi Inscriptional	Pahlavi Psalter
Pahlavi Inscriptional	Parthian Inscriptional
Pahlavi Psalter	Parthian Inscriptional
Old Sogdian	Sogdian
Hiragana	Katakana
Old Turkic Orkhon	Old Turkic Yenisei
Meroitic Hieroglyphs	Meroitic Cursive
Tai Le	Tai Viet
Tai Tham	Tai Le

Tai Tham	Tai Viet
Tagalog	Hanunoo
Tagalog	Buhid
Tagalog	Tagbanwa
Hanunoo	Buhid
Hanunoo	Tagbanwa
Buhid	Tagbanwa
CJK	Kanbun
Devanagari	Nandinagari
Devanagari	Kaithi
Devanagari	Modi

Table 7: Level 2: similar script pairs.

Script A	**Script B**
Greek	Cyrillic
Greek	Glagolitic
Coptic	Cyrillic
Greek	Coptic
Greek	Gothic
Latin	Greek
Latin	Coptic
Latin	Old Italic
Greek	Lycian
Greek	Lydian
Greek	Carian
Linear B	Greek
Cypriot Syllabary	Greek
Phoenician	Greek
Phoenician	Latin
Phoenician	Hebrew
Phoenician	Imperial Aramaic
Imperial Aramaic	Hebrew
Imperial Aramaic	Nabataean
Imperial Aramaic	Syriac
Imperial Aramaic	Samaritan
Imperial Aramaic	Mandaic
Hebrew	Syriac
Hebrew	Samaritan
Syriac	Mandaic
Nabataean	Arabic
Palmyrene	Arabic
Imperial Aramaic	Arabic
Syriac	Arabic
Avestan	Old Persian
Avestan	Pahlavi Inscriptional
Avestan	Sogdian
Old Persian	Pahlavi Inscriptional
Pahlavi Inscriptional	Sogdian
Manichaean	Sogdian
Old South Arabian	Ge‘ez
Brahmi	Devanagari
Brahmi	Bengali
Brahmi	Gurmukhi
Devanagari	Bengali
Devanagari	Gujarati
Devanagari	Gurmukhi
Devanagari	Oriya

Bengali	Gujarati
Bengali	Gurmukhi
Bengali	Oriya
Bengali	Syloti Nagri
Gujarati	Gurmukhi
Gujarati	Oriya
Devanagari	Sharada
Devanagari	Takri
Sharada	Takri
Khojki	Khudawadi
Devanagari	Siddham
Devanagari	Bhaiksuki
Devanagari	Tirhuta
Newa	Bhaiksuki
Grantha	Tamil
Grantha	Malayalam
Grantha	Kannada
Grantha	Telugu
Tamil	Malayalam
Kannada	Telugu
Malayalam	Sinhala
Myanmar	Thai
Myanmar	Khmer
Thai	Lao
Thai	Khmer
Lao	Khmer
Khmer	Cham
Javanese	Balinese
Javanese	Sundanese
Balinese	Buginese
Sundanese	Rejang
Batak	Rejang
Tibetan	Phags-pa
Tibetan	Lepcha
Tibetan	Limbu
Tibetan	Marchen
Phags-pa	Mongolian
Tibetan	Soyombo
Tibetan	Zanabazar Square
Soyombo	Zanabazar Square
Ahom	Chakma
Mongolian	Old Uyghur
Old Uyghur	Old Turkic Orkhon
Linear A	Linear B
Linear A	Cypro-Minoan
Cypro-Minoan	Cypriot Syllabary
Latin	Cyrillic
Latin	Glagolitic
Old Italic	Greek
Cuneiform	Old Persian
Latin	Armenian
Latin	Georgian Asomtavruli
Latin	Georgian Mkhedruli
Greek	Armenian
Greek	Georgian Asomtavruli
Greek	Georgian Mkhedruli
Cyrillic	Armenian
Cyrillic	Georgian Asomtavruli
Glagolitic	Armenian

Latin	Hebrew
Latin	Arabic
Latin	Imperial Aramaic
Greek	Hebrew
Greek	Arabic
Greek	Imperial Aramaic
Armenian	Hebrew
Armenian	Arabic
Georgian Asomtavruli	Hebrew
Georgian Mkhedruli	Arabic
Ge‘ez	Hebrew
Ge‘ez	Arabic
Ge‘ez	Imperial Aramaic
Devanagari	Tamil
Devanagari	Grantha
Devanagari	Kannada
Bengali	Tamil
Gujarati	Malayalam
Gurmukhi	Telugu
Oriya	Sinhala
Devanagari	Thai
Devanagari	Myanmar
Bengali	Khmer
Gujarati	Lao
Tibetan	Myanmar
Tibetan	Thai
Tamil	Thai
Tamil	Khmer
Grantha	Myanmar
Malayalam	Khmer
Devanagari	Javanese
Bengali	Balinese
Gujarati	Sundanese
Tamil	Javanese
Grantha	Balinese
Devanagari	Tibetan
Bengali	Tibetan
Brahmi	Tibetan
Avestan	Imperial Aramaic
Pahlavi Inscriptional	Syriac

Table 8: Level 3: very different script pairs.

Script A	**Script B**
Latin	Hangul Jamo
Greek	Hangul Jamo
Cyrillic	Hangul Jamo
Arabic	Hangul Jamo
Hebrew	Hangul Jamo
Armenian	Hangul Jamo
Latin	CJK
Greek	CJK
Cyrillic	CJK
Arabic	CJK
Hebrew	CJK
Latin	Tangut
Greek	Tangut
Latin	Hiragana

Latin	Katakana
Greek	Hiragana
Greek	Katakana
Cyrillic	Hiragana
Arabic	Katakana
Hebrew	Hiragana
Imperial Aramaic	Hangul Jamo
Syriac	Hangul Jamo
Nabataean	Hangul Jamo
Imperial Aramaic	CJK
Syriac	Tangut
Arabic	Hiragana
Hebrew	Katakana
Imperial Aramaic	Hiragana
Devanagari	Hangul Jamo
Bengali	Hangul Jamo
Tamil	Hangul Jamo
Thai	Hangul Jamo
Javanese	Hangul Jamo
Tibetan	Hangul Jamo
Devanagari	CJK
Bengali	CJK
Tamil	CJK
Thai	CJK
Javanese	Tangut
Tibetan	CJK
Devanagari	Hiragana
Bengali	Katakana
Tamil	Hiragana
Thai	Katakana
Javanese	Hiragana
Tibetan	Katakana
Hangul Jamo	CJK
Hangul Jamo	Tangut
Hangul Jamo	Hiragana
Hangul Jamo	Katakana
CJK	Hiragana
CJK	Katakana
Tangut	Hiragana
Mongolian	Latin
Mongolian	Greek
Mongolian	Arabic
Old Uyghur	Latin
Old Turkic Orkhon	Greek
Mongolian	Devanagari
Mongolian	Bengali
Mongolian	Tamil
Old Uyghur	Thai
Old Turkic Orkhon	Tibetan
Mongolian	CJK
Mongolian	Hangul Jamo
Mongolian	Hiragana
Old Uyghur	Katakana
Egyptian Hieroglyphs	Latin
Egyptian Hieroglyphs	Greek
Egyptian Hieroglyphs	Arabic
Meroitic Hieroglyphs	Hebrew
Egyptian Hieroglyphs	Devanagari
Egyptian Hieroglyphs	Tamil

Meroitic Cursive	Bengali
Egyptian Hieroglyphs	CJK
Egyptian Hieroglyphs	Hangul Jamo
Egyptian Hieroglyphs	Hiragana
Meroitic Hieroglyphs	Katakana
Cuneiform	Latin
Cuneiform	Greek
Cuneiform	Arabic
Cuneiform	Devanagari
Cuneiform	Tamil
Cuneiform	Thai
Cuneiform	CJK
Cuneiform	Hangul Jamo
Cuneiform	Hiragana
Anatolian Hieroglyphs	Latin
Anatolian Hieroglyphs	Greek
Anatolian Hieroglyphs	Arabic
Anatolian Hieroglyphs	Devanagari
Anatolian Hieroglyphs	CJK
Anatolian Hieroglyphs	Hangul Jamo
Tifinagh	Latin
Tifinagh	Greek
Tifinagh	Arabic
Tifinagh	Devanagari
Tifinagh	CJK
Tifinagh	Hangul Jamo
Tifinagh	Hiragana
Old Hungarian	Latin
Old Hungarian	Greek
Old Hungarian	Arabic
Old Hungarian	Devanagari
Old Hungarian	CJK
Old Hungarian	Hangul Jamo
Phaistos Disc	Latin
Phaistos Disc	Greek
Phaistos Disc	Arabic
Phaistos Disc	Devanagari
Phaistos Disc	CJK
Phaistos Disc	Hangul Jamo
Todhri	Latin
Todhri	Greek
Todhri	Arabic
Todhri	Devanagari
Todhri	CJK
Todhri	Hangul Jamo
Elder Futhark	Hangul Jamo
Elder Futhark	CJK
Elder Futhark	Hiragana
Elder Futhark	Devanagari
Ogham	Hangul Jamo
Ogham	CJK
Ogham	Devanagari
Latin	Devanagari
Latin	Bengali
Latin	Tamil
Greek	Devanagari
Greek	Tamil
Cyrillic	Devanagari
Arabic	Devanagari

Arabic	Tamil
Hebrew	Bengali
Armenian	Devanagari
Linear A	Hangul Jamo
Linear A	CJK
Linear A	Devanagari
Linear A	Arabic
Linear B	Hangul Jamo
Linear B	CJK
Linear B	Devanagari

B Implementation Details

B.1 Backbone Architectures

This subsection provides implementation details for the backbone encoders used in our experiments. As described in Section 4.1, we evaluate a lightweight custom CNN, a Siamese-style convolutional encoder inspired by Koch et al. (2015), and three ResNet variants (He et al., 2016). In all cases, the backbone outputs a fixed-dimensional glyph embedding subsequently used for supervised contrastive learning, self-supervised adaptation, or retrieval-based evaluation. Unless otherwise stated, input images are grayscale glyph images of size 105×105 pixels and embeddings are ℓ_2-normalized before similarity computation in the embedding space.

Simple CNN. Our *Simple CNN* backbone is a compact convolutional encoder specifically designed for glyph images. It consists of three convolutional blocks, each composed of a convolution, batch normalization, ReLU activation, and 2×2 max-pooling. The successive channel dimensions are $32 \to 64 \to 128$, with all convolutions using 3×3 kernels and padding of 1. After the final convolutional block, the feature map is reduced by adaptive average pooling to a 1×1 spatial representation, flattened, and passed through a two-layer MLP head with dimensions $128 \to 256 \to d$, where $d \in \{64, 128, 256\}$ denotes the embedding dimension. The final embedding is ℓ_2-normalized.

Formally, the architecture is:

$$\begin{aligned}&\mathrm{Conv}(1, 32, 3) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)\\ &\to \mathrm{Conv}(32, 64, 3) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)\\ &\to \mathrm{Conv}(64, 128, 3) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)\\ &\to \mathrm{AdaptiveAvgPool}(1, 1) \to \mathrm{FC}(128, 256) \to \mathrm{ReLU} \to \mathrm{FC}(256, d).\end{aligned}$$

For methods requiring an auxiliary projection head during training, the implementation also supports an optional projection MLP of size $d \to 256 \to d$, although in our hybrid BYOL adaptation the predictor operates directly on backbone embeddings.

Siamese CNN. Our *Siamese-like CNN* follows the convolutional branch introduced by Koch et al. (2015) for one-shot character recognition. In our setting, we use the encoder branch as a standalone backbone producing glyph embeddings rather than directly optimizing the original pairwise verification objective. The network contains four convolutional layers with progressively increasing numbers of channels: $64 \to 128 \to 128 \to 256$. The corresponding kernel sizes are 10×10, 7×7, 4×4, and 4×4. Batch normalization and ReLU are applied after each convolution, and max-pooling is used after the first three convolutional layers.

After the convolutional stack, the resulting feature maps are pooled by adaptive average pooling to a 1×1 representation, flattened, and projected through a fully connected layer to a 4096-dimensional embedding. Following the original Koch-style formulation, a sigmoid activation is applied to the output embedding. In contrast to the other backbones, the ℓ_2 normalization is therefore applied outside the network, before similarity computation.

The architecture can be summarized as:

$$\mathrm{Conv}(1, 64, 10) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)$$

$$\to \mathrm{Conv}(64, 128, 7) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)$$
$$\to \mathrm{Conv}(128, 128, 4) \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{MaxPool}(2)$$
$$\to \mathrm{Conv}(128, 256, 4) \to \mathrm{BN} \to \mathrm{ReLU}$$
$$\to \mathrm{AdaptiveAvgPool}(1, 1) \to \mathrm{FC}(256, 4096) \to \sigma.$$

This architecture is substantially higher-dimensional than the other backbones and was included to test whether a strong one-shot learning architecture remains effective when repurposed for contrastive and self-supervised glyph representation learning.

ResNets. We evaluate three standard residual networks (He et al., 2016): ResNet-18, ResNet-34, and ResNet-50. These models are adapted to grayscale glyph images by replacing the first convolutional layer with a single-channel input convolution:

$$\mathrm{Conv}(1, 64, 7 \times 7, \mathrm{stride} = 2, \mathrm{padding} = 3).$$

The remainder of the ResNet architecture is unchanged. The original classification head is replaced with a linear layer projecting the final pooled representation to an embedding of dimension $d \in \{64, 128, 256\}$. The final output is then ℓ_2-normalized.

When pretrained initialization is enabled, the implementation uses the default torchvision weights for the corresponding ResNet variant. In the experiments reported in Section 4, the ResNet backbones provide a family of increasingly expressive convolutional encoders, allowing us to assess how the proposed two-stage training strategy behaves as model capacity grows from ResNet-18 to ResNet-50.

DINOv2-ViT-S/14. In addition to convolutional encoders trained on glyph data, we evaluate DINOv2-ViT-S/14 as a large pretrained visual baseline, both in frozen form and with a lightweight fine-tuning strategy on the target glyph data. As noted in Section 4.1, this model contains approximately 22M parameters and serves as a reference to test whether generic foundation-model visual features transfer to ancient and historical scripts without domain-specific pretraining. Since DINOv2 is used as an external pretrained baseline rather than implemented in our custom backbone module, we refer the reader to the original DINOv2 work for its full architectural specification.

Normalization. Every embedding is normalized to unit ℓ_2 norm before similarity computation, which makes cosine similarity equivalent to the dot product and ensures a consistent geometry for supervised contrastive learning, BYOL adaptation, and nearest-neighbor retrieval. For the Simple CNN and all ResNet variants, this normalization is the last operation of the network; the Siamese CNN keeps the original sigmoid embedding of Koch et al. (2015) and is normalized outside the network.

B.2 Hyperparameters

Hyperparameters were optimized independently for each method and backbone with Optuna (one study of 30 trials per configuration, maximizing 1-NN validation accuracy). Tables 9–11 report, for each study, the best trial and its objective value together with the selected hyperparameters. "–" denotes a hyperparameter that does not apply to the backbone.

Table 9: Best hyperparameters of the hybrid model (Stage 1 teacher and Stage 2 student). No projection head is used, and no learning-rate scheduler is applied to the teacher. †Reported by the study as a single backbone/main learning rate and weight decay.

	Simple CNN	Siamese CNN	ResNet-18	ResNet-34	ResNet-50
Best trial (score)	26 (0.9614)	19 (0.9714)	15 (0.9900)	25 (0.9886)	3 (0.9998)
ImageNet-pretrained	–	–	yes	yes	no
Batch size	128	128	512	512	128
Embedding dimension	256	4096	128	64	128
Temperature τ	0.0984	0.2228	0.1101	0.0545	0.0804
Teacher epochs	96	155	159	132	192
Teacher LR	1.51×10^{-4}	6.53×10^{-4}	5.02×10^{-4}	4.16×10^{-4}	3.34×10^{-4}
Teacher WD	1.19×10^{-7}	2.84×10^{-6}	1.49×10^{-5}	8.99×10^{-6}	1.91×10^{-7}
Student epochs	475	195	173	339	193
Student LR	8.53×10^{-5}	1.49×10^{-4}	3.01×10^{-5}	1.04×10^{-4}	$6.24\times10^{-5\dagger}$
Student WD	4.07×10^{-6}	2.41×10^{-7}	3.77×10^{-7}	2.33×10^{-6}	$1.75\times10^{-6\dagger}$
EMA decay κ	0.9944	0.9600	0.9820	0.9992	0.9830
Predictor hidden dim.	256	1024	256	512	512
Predictor LR multiplier	3.28	2.29	3.25	1.07	1.55
Warmup epochs	19	45	30	53	8
Gradient clipping	0.0413	0.1278	0.5987	0.695	1.385

Table 10: Best hyperparameters of the BYOL baseline. No projection head is used. ResNet-34 and ResNet-50 share the same configuration: in both studies, the best trial is the first one, drawn at random with the same sampler seed.

	Simple CNN	Siamese CNN	ResNet-18	ResNet-34	ResNet-50
Best trial (score)	26 (0.8043)	25 (0.9600)	16 (1.0)	0 (1.0)	0 (0.9954)
ImageNet-pretrained	–	–	yes	yes	yes
Batch size	128	128	128	256	256
Embedding dimension	64	4096	256	256	256
Epochs	292	272	296	306	306
LR	1.49×10^{-4}	2.95×10^{-4}	2.94×10^{-4}	2.76×10^{-4}	2.76×10^{-4}
WD	2.48×10^{-6}	1.12×10^{-6}	2.61×10^{-6}	1.41×10^{-6}	1.41×10^{-6}
EMA decay κ	0.9638	0.9555	0.9931	0.9958	0.9958
Predictor hidden dim.	1024	1024	512	512	512
Predictor LR multiplier	3.60	3.08	3.98	3.50	3.50
Warmup epochs	34	6	45	49	49
Gradient clipping	0.162	0.0036	1.968	1.058	1.058

Table 11: Best hyperparameters of the Barlow Twins baseline.

	Simple CNN	Siamese CNN	ResNet-18	ResNet-34	ResNet-50
Best trial (score)	26 (0.9825)	18 (0.9995)	9 (1.0)	16 (1.0)	11 (0.9970)
ImageNet-pretrained	–	–	yes	no	yes
Batch size	128	128	256	128	512
Embedding dimension	128	4096	64	128	64
Epochs	427	499	496	448	493
LR	1.70×10^{-4}	2.20×10^{-4}	1.43×10^{-4}	2.14×10^{-4}	9.29×10^{-5}
WD	3.43×10^{-7}	4.64×10^{-7}	2.30×10^{-6}	2.70×10^{-6}	2.41×10^{-6}
$\lambda_{\text{off-diag}}$	0.00180	0.00600	0.000447	0.00415	0.0415
Projector output dim.	512	1024	4096	512	4096
Projector hidden dim.	1024	512	2048	1024	1024
Projector LR multiplier	2.58	3.74	3.49	2.42	2.89
Warmup epochs	16	59	35	48	34
Gradient clipping	1.936	0.0519	1.793	1.396	1.902

C Additional Results

C.1 Stage-2-Only BYOL and Fully Fine-Tuned DINOv2

This subsection reports two additional experiments that complement the main benchmark (Section 4): a BYOL baseline trained solely on the Stage-2 data, without the Stage-1 initialization, and a fully fine-tuned DINOv2 baseline. Together they clarify the trade-off between glyph-level retrieval and script-level ranking. Results are summarized in Table 12.

Table 12: Additional experiments. We report BYOL trained only on Stage-2 data, without Stage-1 initialization, and a fully fine-tuned DINOv2 baseline. Results clarify the trade-off between glyph-level retrieval and script-level ranking.

Backbone	Method	N20R1	N20R5	NDCG10	Spearman
Simple CNN	BYOL Stage-2	59.25	85	0.18	0.534
Siamese CNN	BYOL Stage-2	95.5	99.75	0.259	0.6296
ResNet-18	BYOL Stage-2	99.75	100	0.2592	0.578
ResNet-34	BYOL Stage-2	99.75	100	0.2662	0.2982
ResNet-50	BYOL Stage-2	99.50	100	0.2649	0.5741
DINOv2	Fully fine-tuned	82.00	98.75	0.2374	0.6268

BYOL trained on Stage-2 data only. We trained standard BYOL exclusively on the Stage-2 unsupervised set (no invented alphabets). As reported in Table 12, BYOL Stage-2 only achieves higher N20R1/N20R5 on the ResNet backbones, but its NDCG@10 is lower than our method on four of the five backbones, the only exception being ResNet-18 (0.2592 vs. 0.2532), with the largest gap on ResNet-50 (0.2649 vs. 0.3178). This supports our central claim: without the semantic prior from Stage 1, the model tends to overfit to local discriminative features and recovers a weaker script-level geometry. The strong glyph-level accuracy of BYOL Stage-2 illustrates the trade-off our framework targets, since pushing all character classes apart, including historically related ones, improves local discrimination but not script-level ranking.

DINOv2 fully fine-tuned. We ran DINOv2-ViT-S/14 with full fine-tuning on the Stage-2 data only, using the same BYOL-style objective and head hyperparameters as the frozen-backbone variant. The results (Table 12) show NDCG@10 $= 0.2374$, still below our method on all backbones. This confirms that large vision transformers pre-trained on natural images do not transfer well to ancient script analysis, regardless of the fine-tuning strategy, underscoring the importance of domain-adapted training.

C.2 t-SNE Visualizations

To complement the quantitative results, we provide qualitative t-SNE visualizations (Van der Maaten & Hinton, 2008) comparing our method with BYOL trained on Stage-2 data only. Figure 4 shows an episode-level visualization of a 20-way 1-shot query.

BYOL Stage-2 can produce tighter local glyph neighbourhoods and solve the 1-shot query, illustrating its strength on local glyph discrimination. Despite these local advantages, BYOL Stage-2 remains worse on NDCG@10 on four of the five backbones, consistent with the trade-off discussed in Section C.1.

C.3 Script Distances per Similarity Level

Figures 5–31 show, for every backbone and training strategy, the distribution of script distances d_s at each ground-truth similarity level. Script distances generally increase with the level, indicating that the embeddings reflect the hierarchy of script relationships. To quantify this effect, we perform *one-sided Mann–Whitney U tests* (Mann & Whitney, 1947) between adjacent levels, testing $1 < 2$ and $2 < 3$. For most methods, the separation between levels 2 and 3 is strongly significant, whereas the separation between levels 1 and 2 is often weaker and frequently non-significant ("ns"). This is

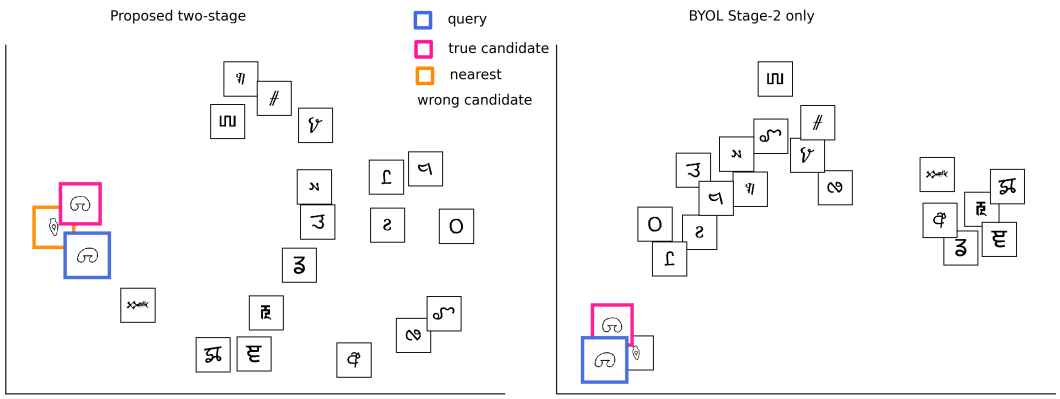


Figure 4: Episode-level visualization of a 20-way 1-shot query. BYOL Stage-2 solves the local glyph query, while our model makes a visually plausible nearest-neighbour error.

particularly visible for several baseline settings, including SupCon, Barlow Twins, frozen DINOv2, and some BYOL configurations. In contrast, our method yields significant ordered comparisons more often, as reflected by the larger number of starred annotations across the plots, suggesting that it more reliably preserves the expected progression of script distances from closer to more distant script families.

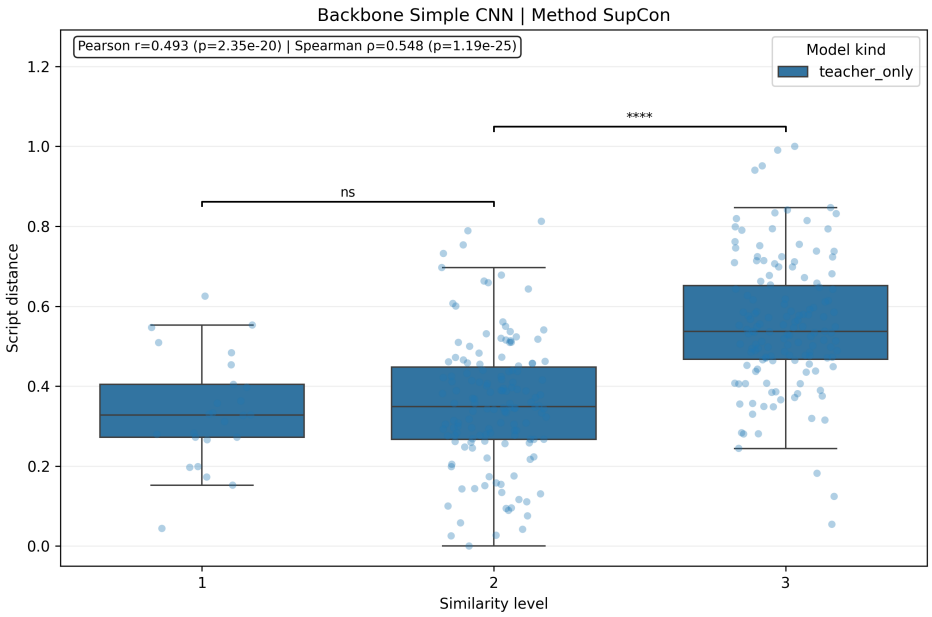


Figure 5: **Backbone: Simple CNN | Method: SupCon.** Script-distance versus similarity-level analysis.

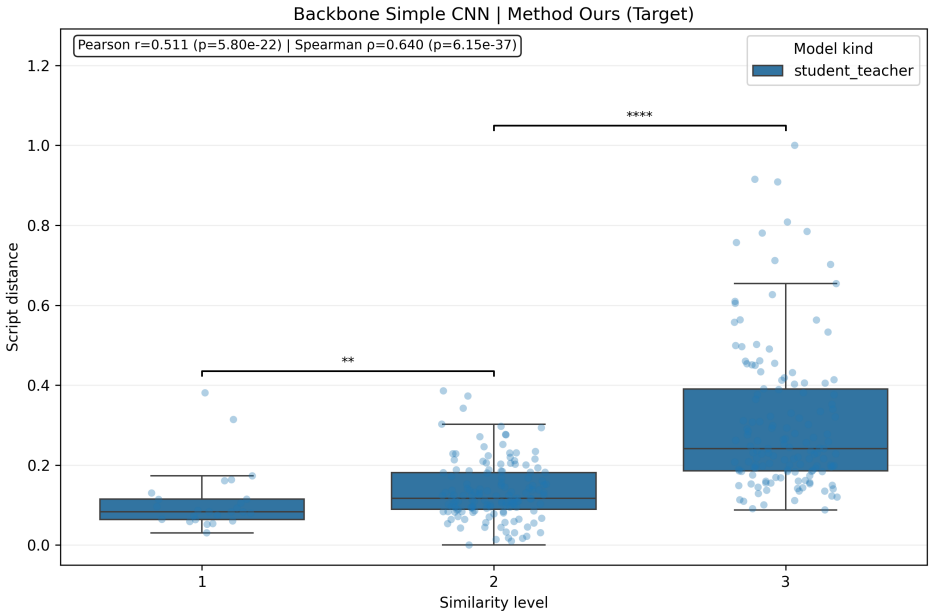


Figure 6: **Backbone: Simple CNN | Method: Ours (Target).** Script-distance versus similarity-level analysis.

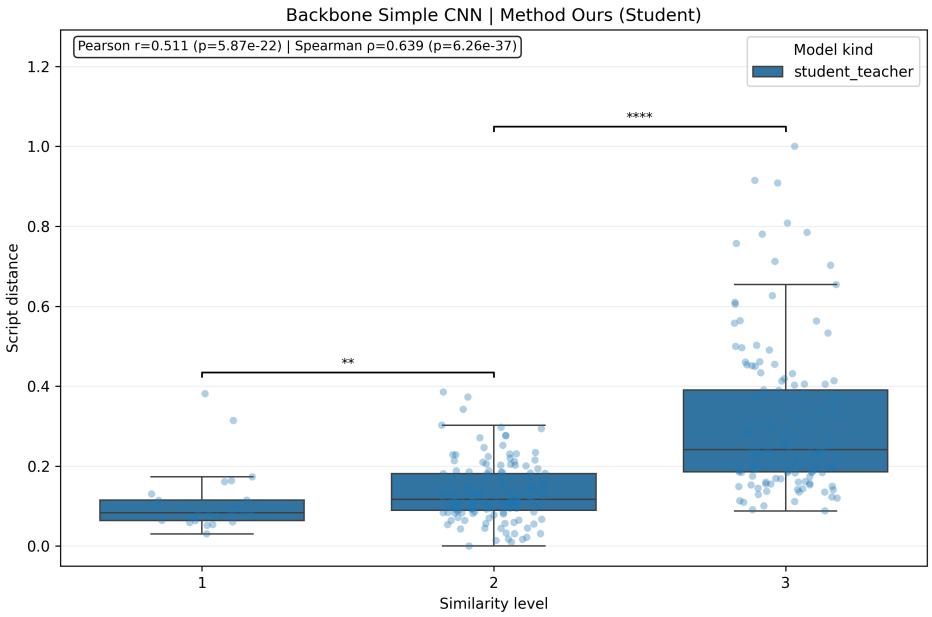


Figure 7: **Backbone: Simple CNN | Method: Ours (Student).** Script-distance versus similarity-level analysis.

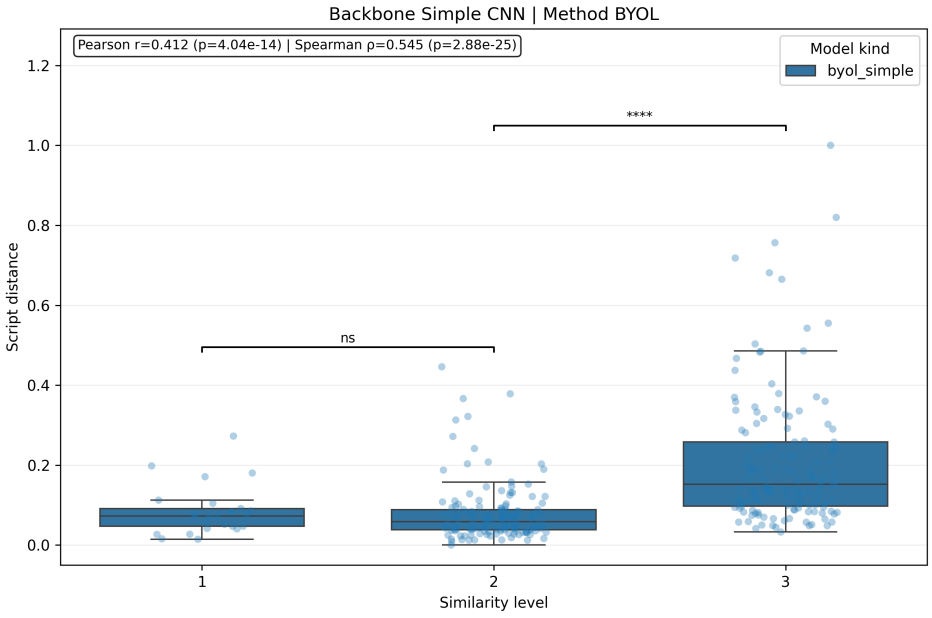


Figure 8: **Backbone: Simple CNN | Method: BYOL.** Script-distance versus similarity-level analysis.

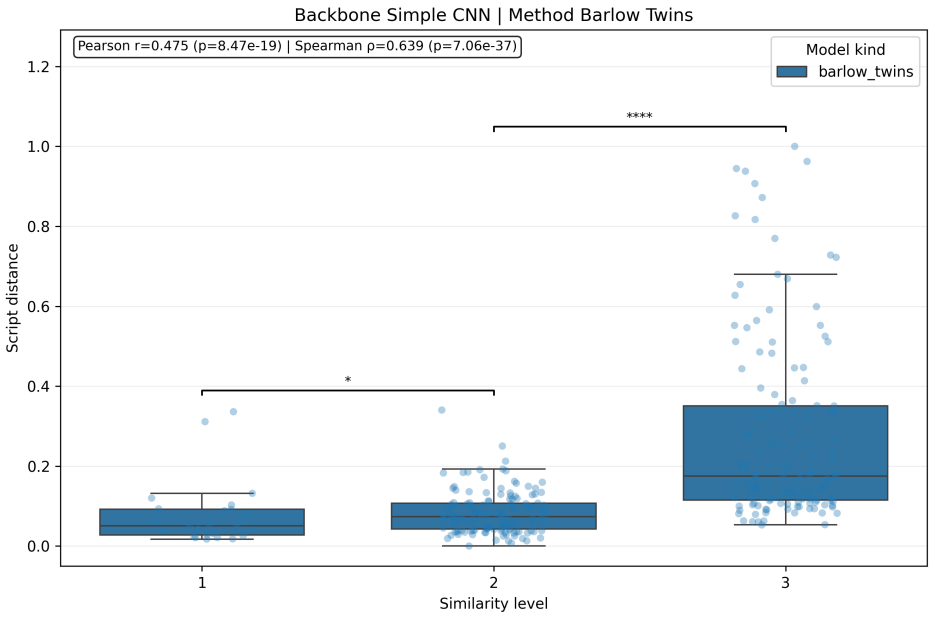


Figure 9: **Backbone: Simple CNN | Method: Barlow Twins.** Script-distance versus similarity-level analysis.

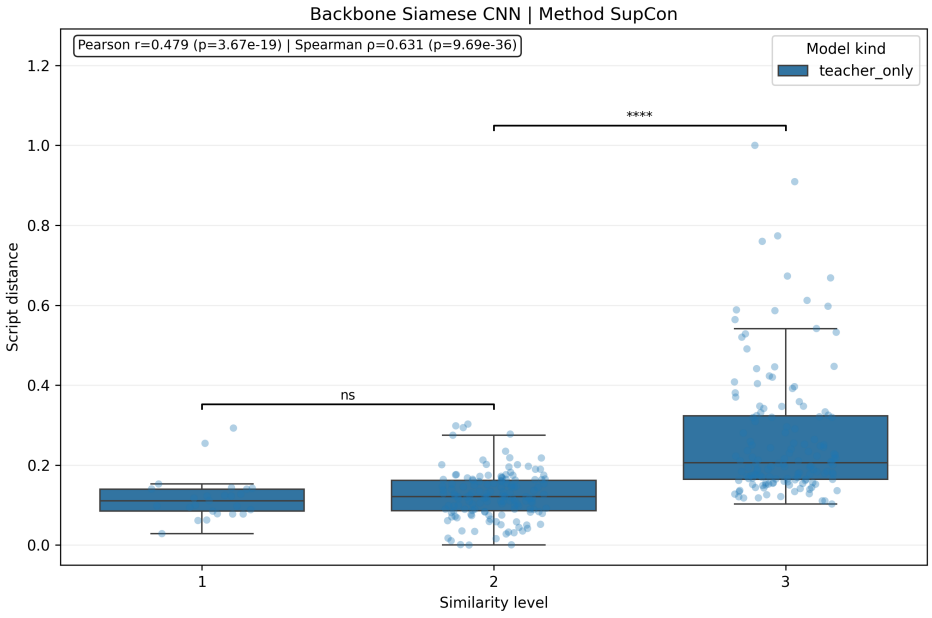


Figure 10: **Backbone: Siamese CNN | Method: SupCon.** Script-distance versus similarity-level analysis.

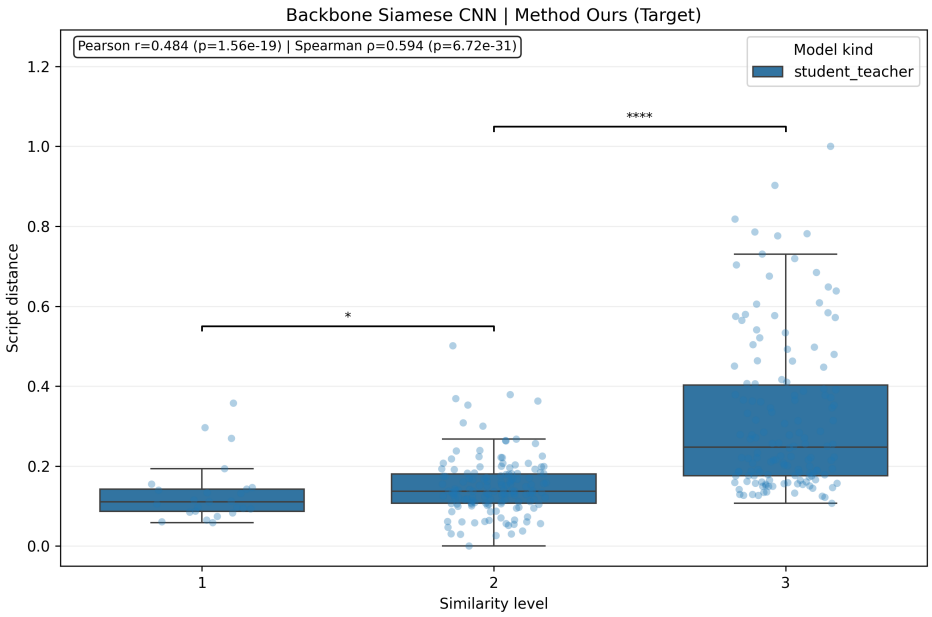


Figure 11: **Backbone: Siamese CNN | Method: Ours (Target).** Script-distance versus similarity-level analysis.

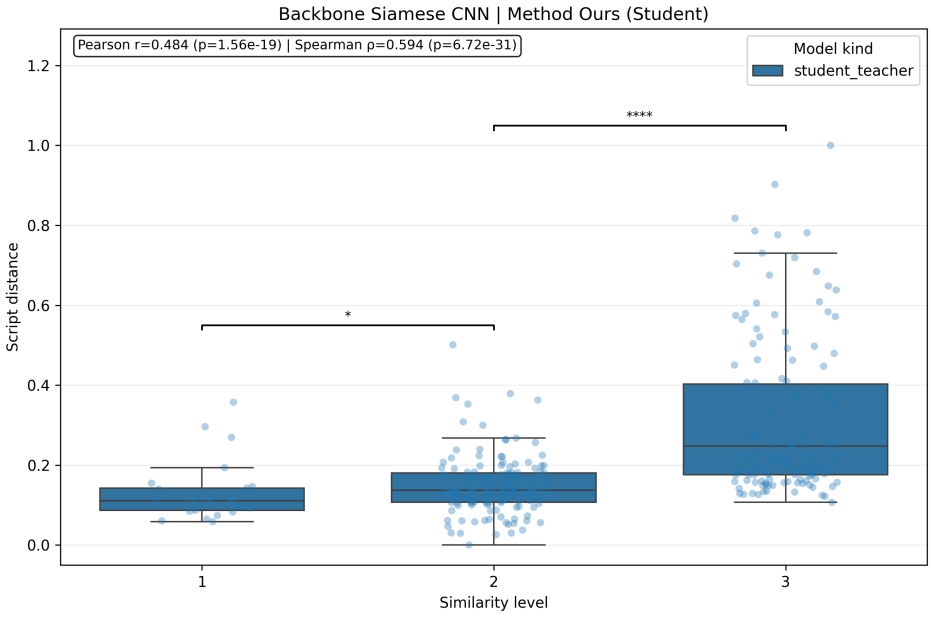


Figure 12: **Backbone: Siamese CNN | Method: Ours (Student).** Script-distance versus similarity-level analysis.

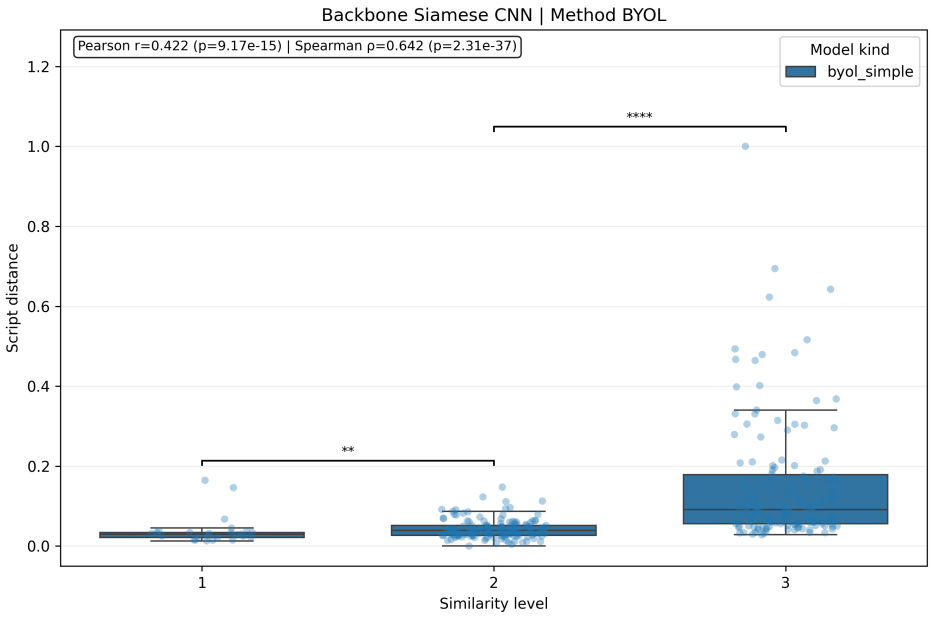


Figure 13: **Backbone: Siamese CNN | Method: BYOL.** Script-distance versus similarity-level analysis.

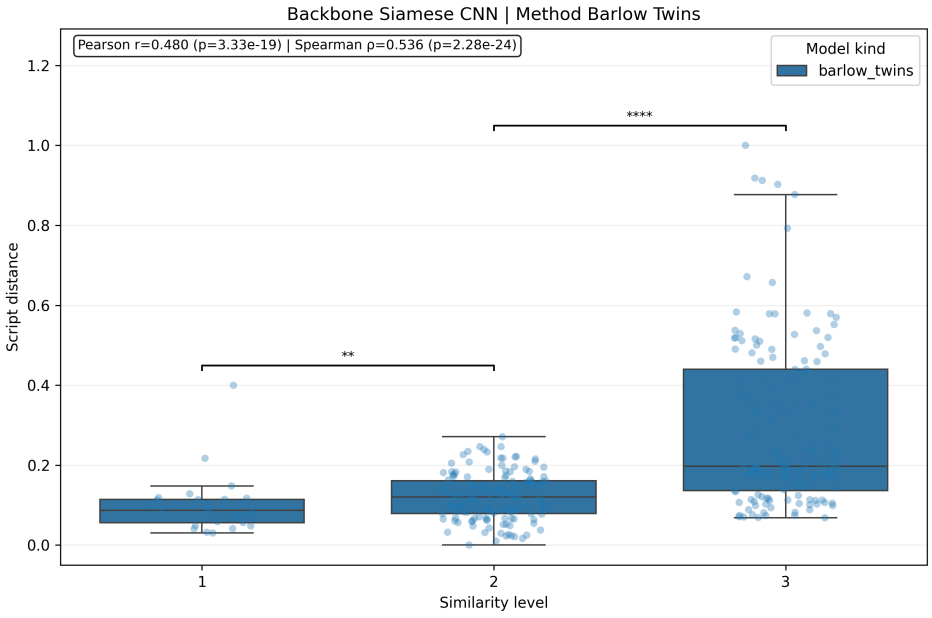


Figure 14: **Backbone: Siamese CNN | Method: Barlow Twins.** Script-distance versus similarity-level analysis.

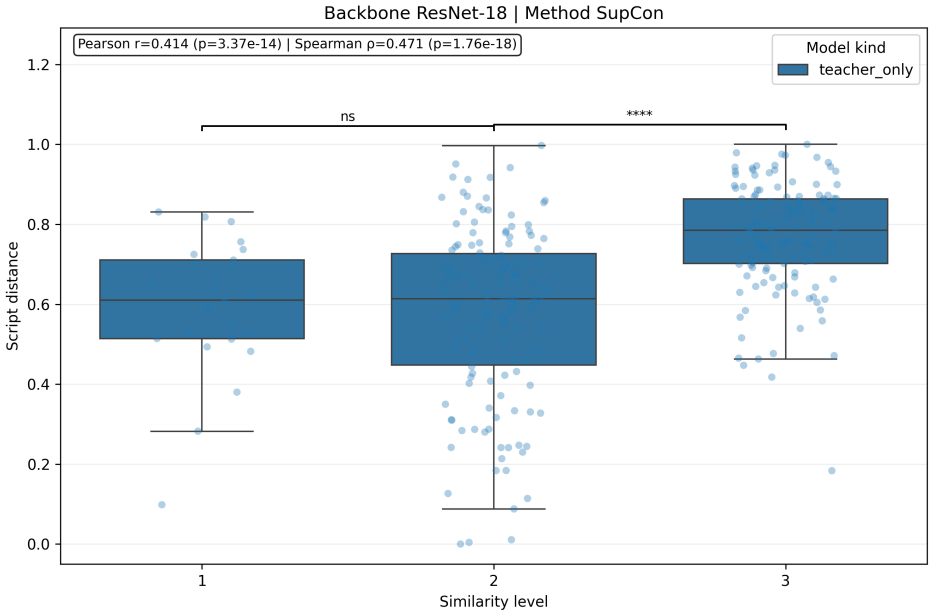


Figure 15: **Backbone: ResNet-18 | Method: SupCon.** Script-distance versus similarity-level analysis.

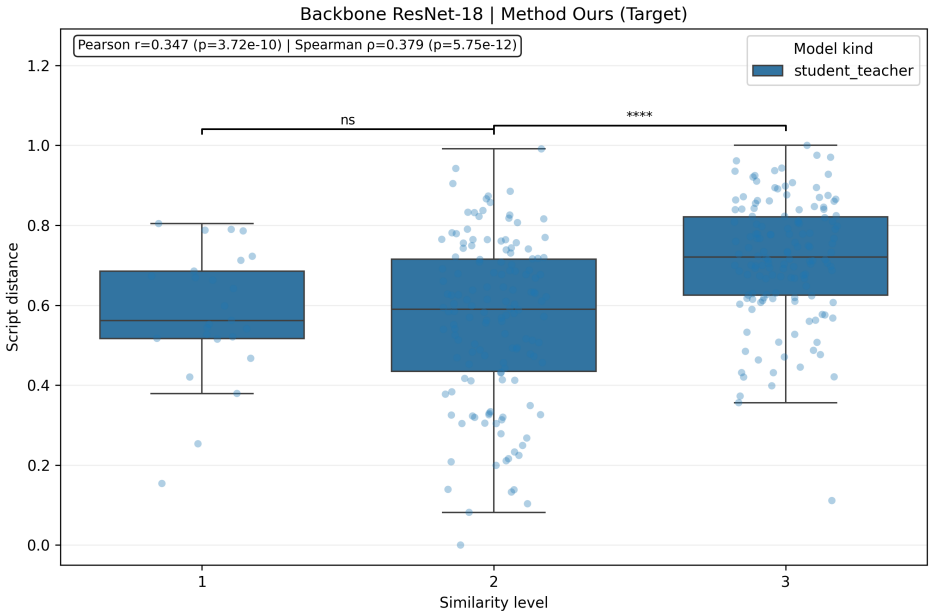


Figure 16: **Backbone: ResNet-18 | Method: Ours (Target).** Script-distance versus similarity-level analysis.

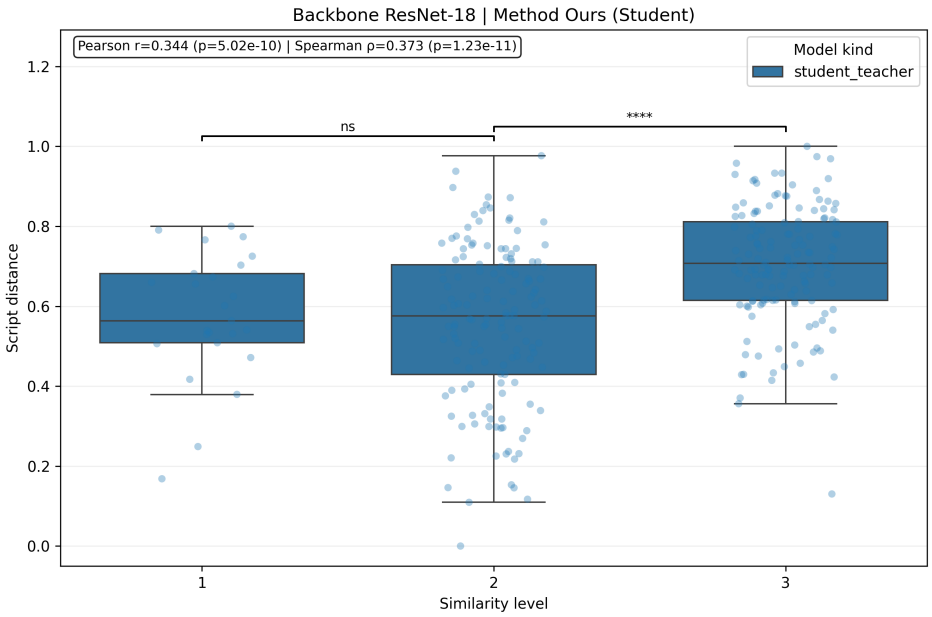


Figure 17: **Backbone: ResNet-18 | Method: Ours (Student).** Script-distance versus similarity-level analysis.

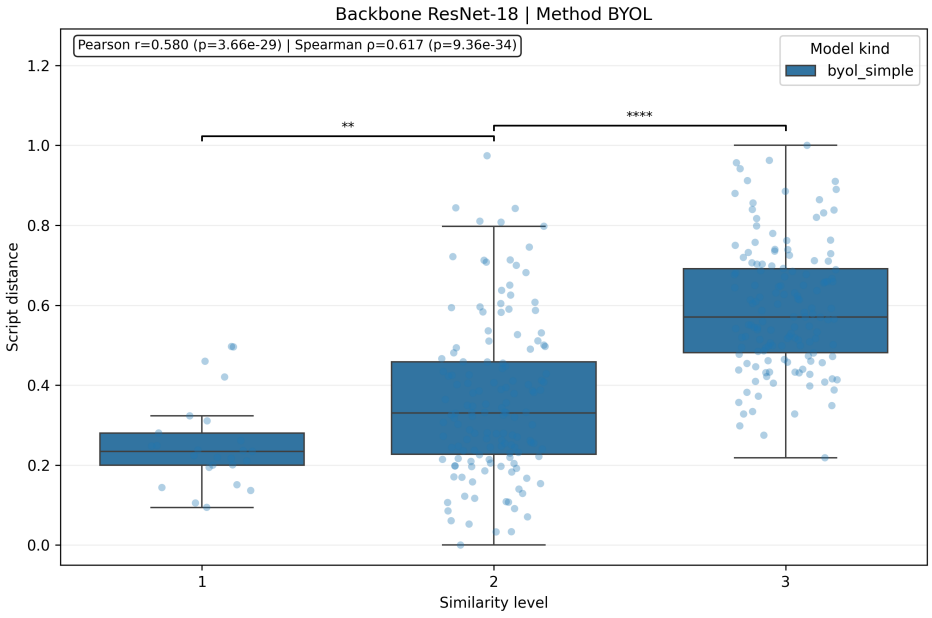


Figure 18: **Backbone: ResNet-18 | Method: BYOL.** Script-distance versus similarity-level analysis.

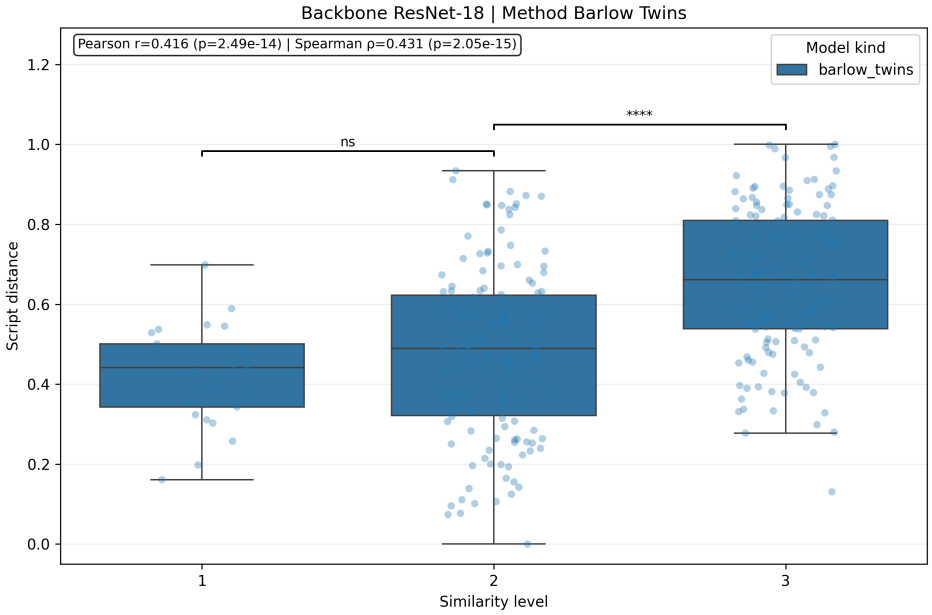


Figure 19: **Backbone: ResNet-18 | Method: Barlow Twins.** Script-distance versus similarity-level analysis.

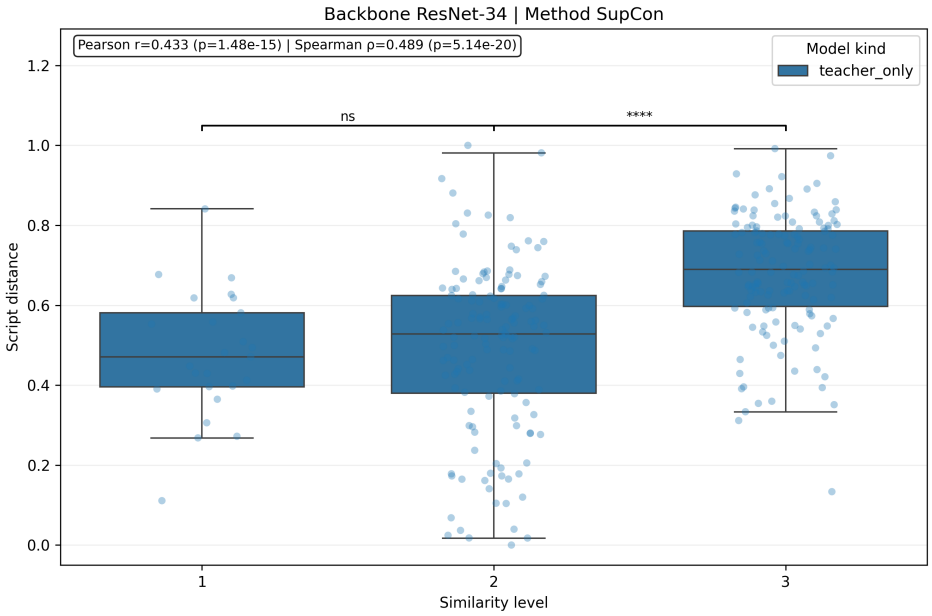


Figure 20: **Backbone: ResNet-34 | Method: SupCon.** Script-distance versus similarity-level analysis.

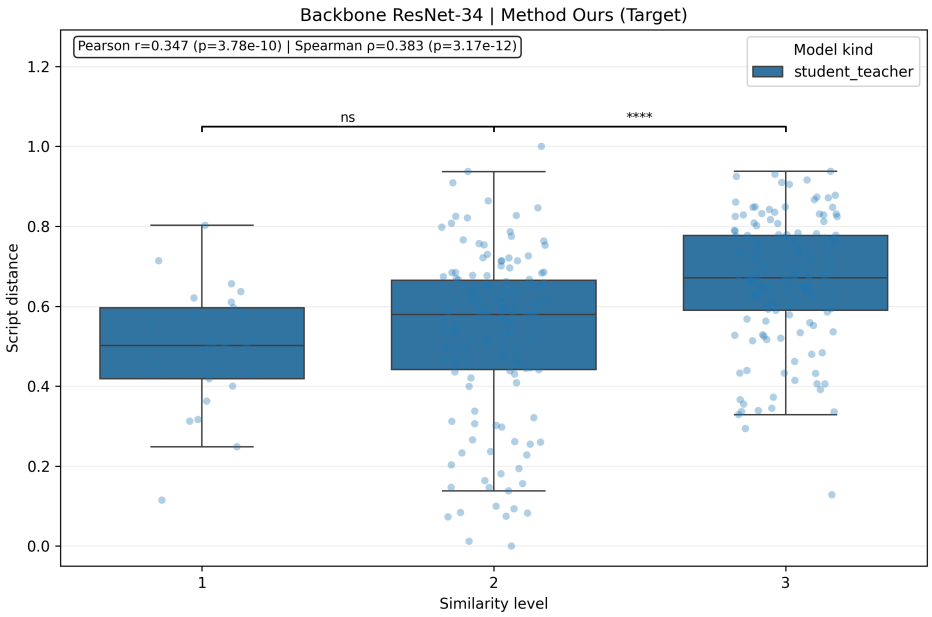


Figure 21: **Backbone: ResNet-34 | Method: Ours (Target).** Script-distance versus similarity-level analysis.

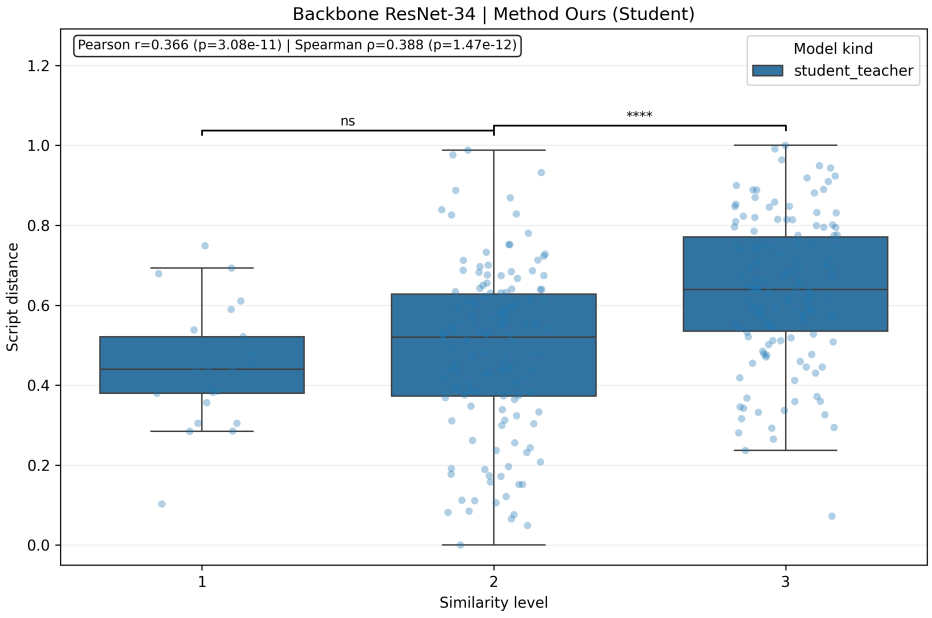


Figure 22: **Backbone: ResNet-34 | Method: Ours (Student).** Script-distance versus similarity-level analysis.

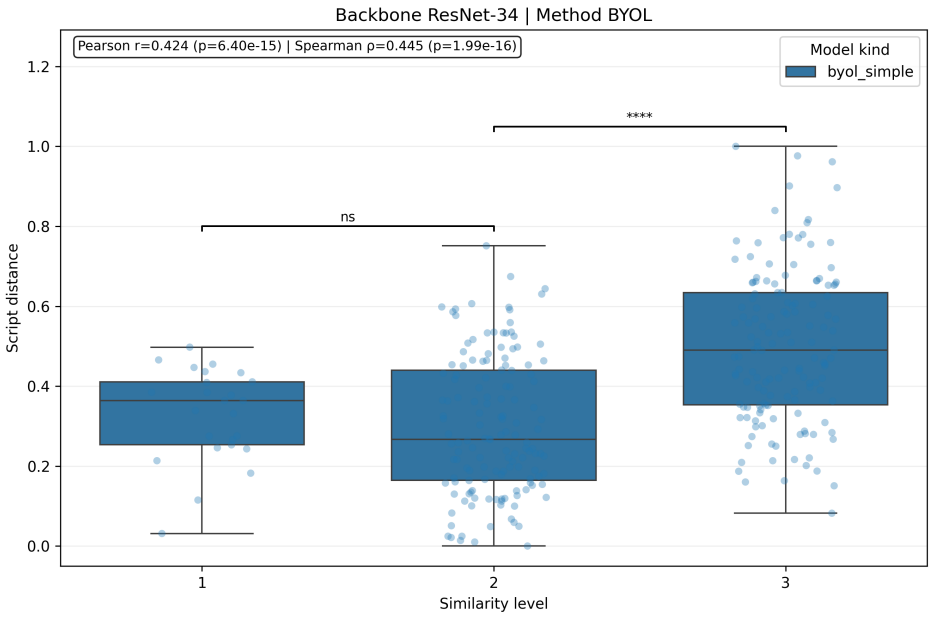


Figure 23: **Backbone: ResNet-34 | Method: BYOL.** Script-distance versus similarity-level analysis.

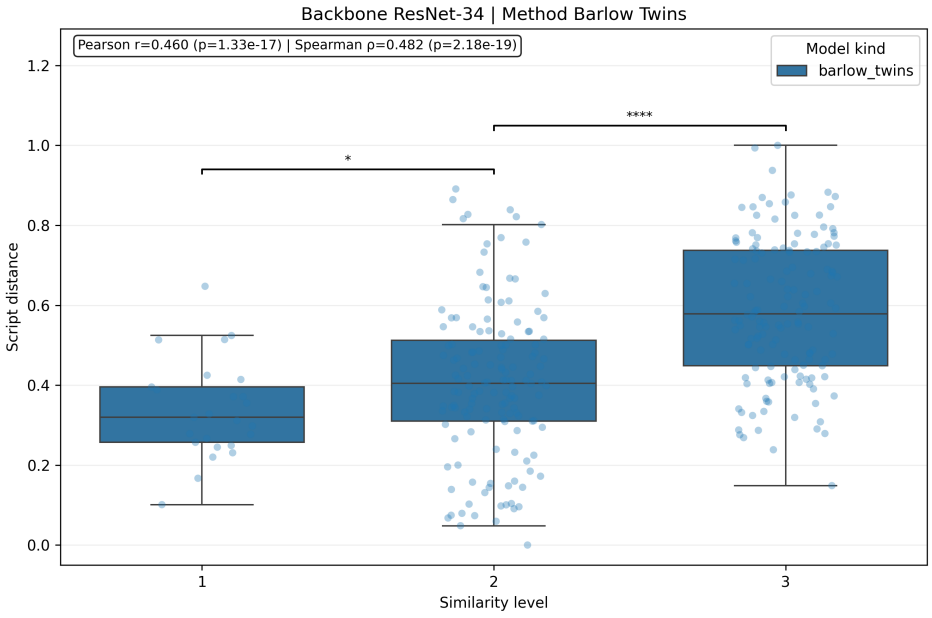


Figure 24: **Backbone: ResNet-34 | Method: Barlow Twins.** Script-distance versus similarity-level analysis.

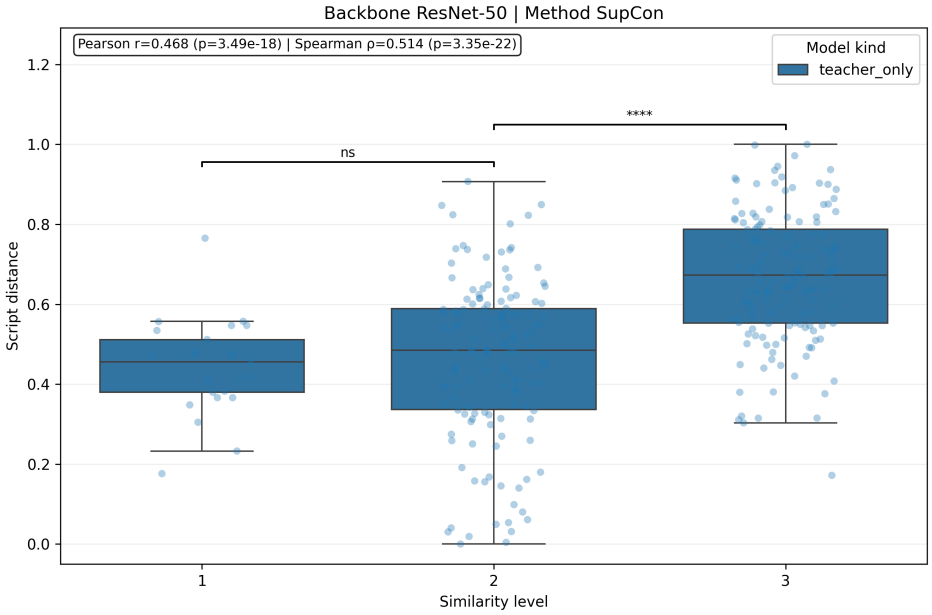


Figure 25: **Backbone: ResNet-50 | Method: SupCon.** Script-distance versus similarity-level analysis.

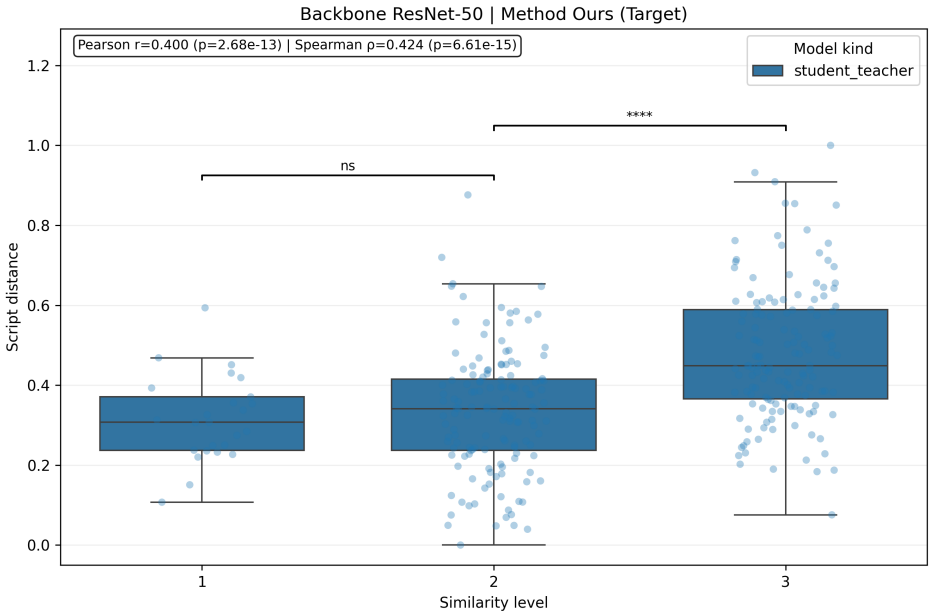


Figure 26: **Backbone: ResNet-50 | Method: Ours (Target).** Script-distance versus similarity-level analysis.

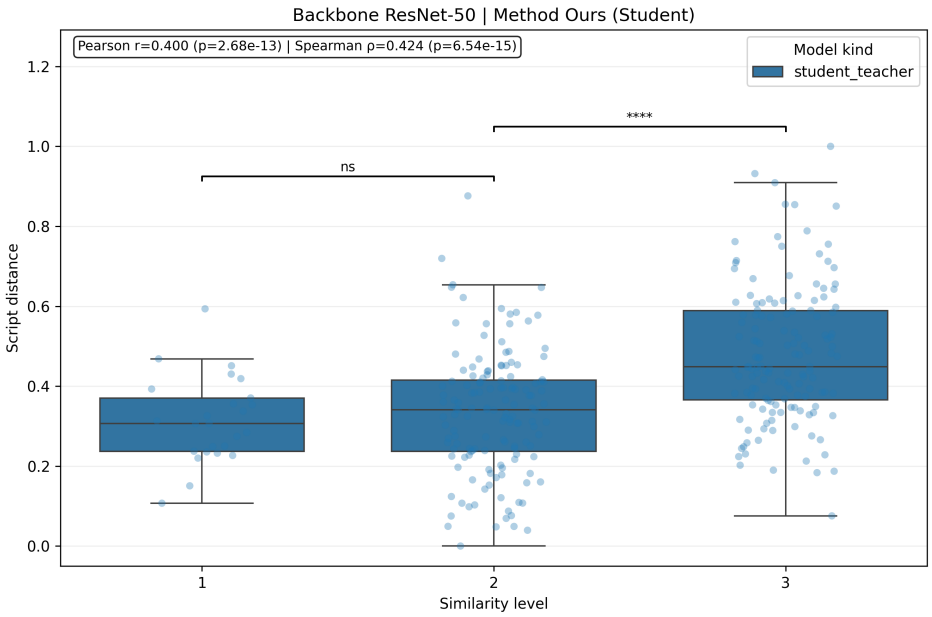


Figure 27: **Backbone: ResNet-50 | Method: Ours (Student).** Script-distance versus similarity-level analysis.

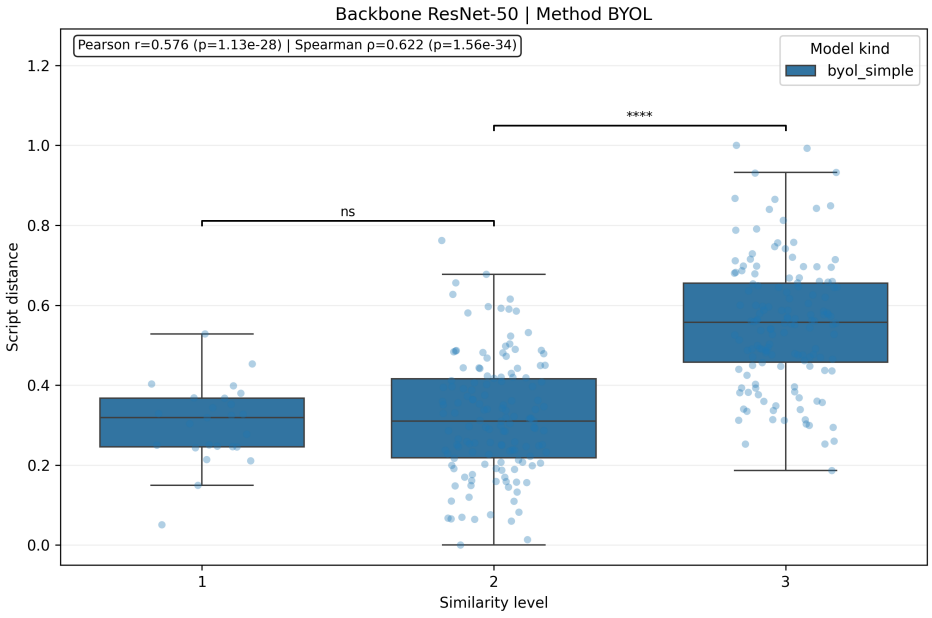


Figure 28: **Backbone: ResNet-50 | Method: BYOL.** Script-distance versus similarity-level analysis.

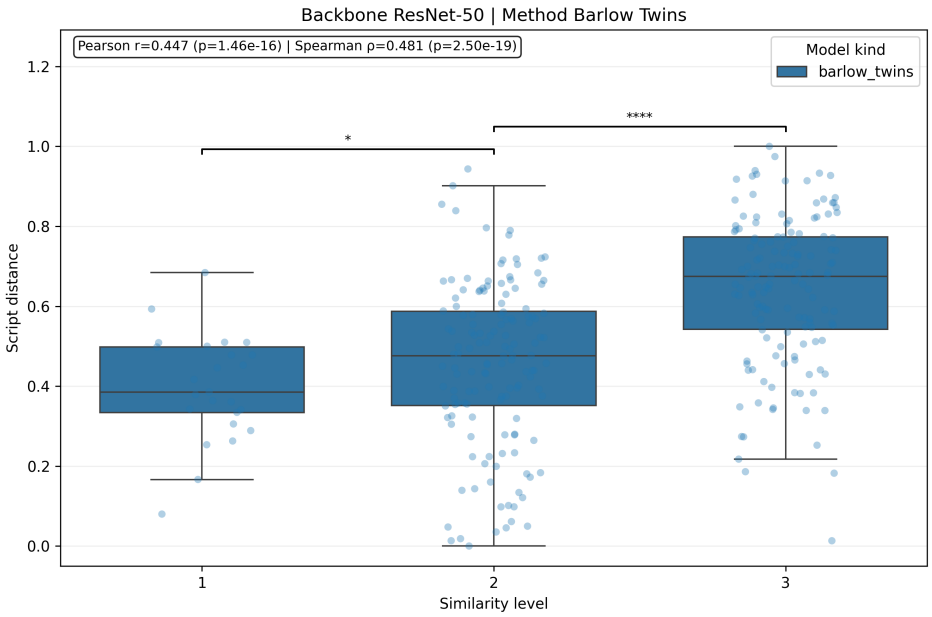


Figure 29: **Backbone: ResNet-50 | Method: Barlow Twins.** Script-distance versus similarity-level analysis.

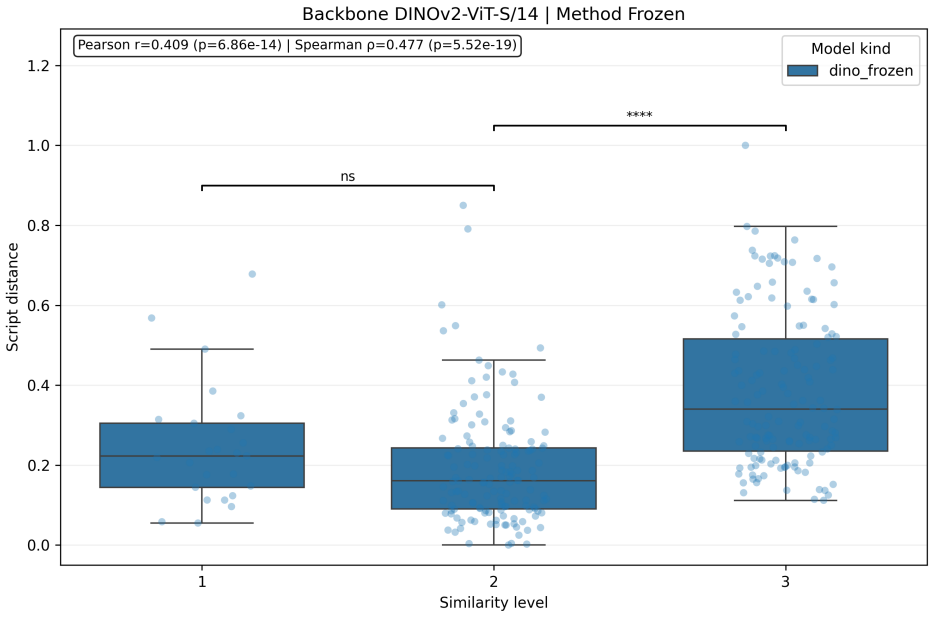


Figure 30: **Backbone: DINOv2-ViT-S/14 | Method: Frozen.** Script-distance versus similarity-level analysis.

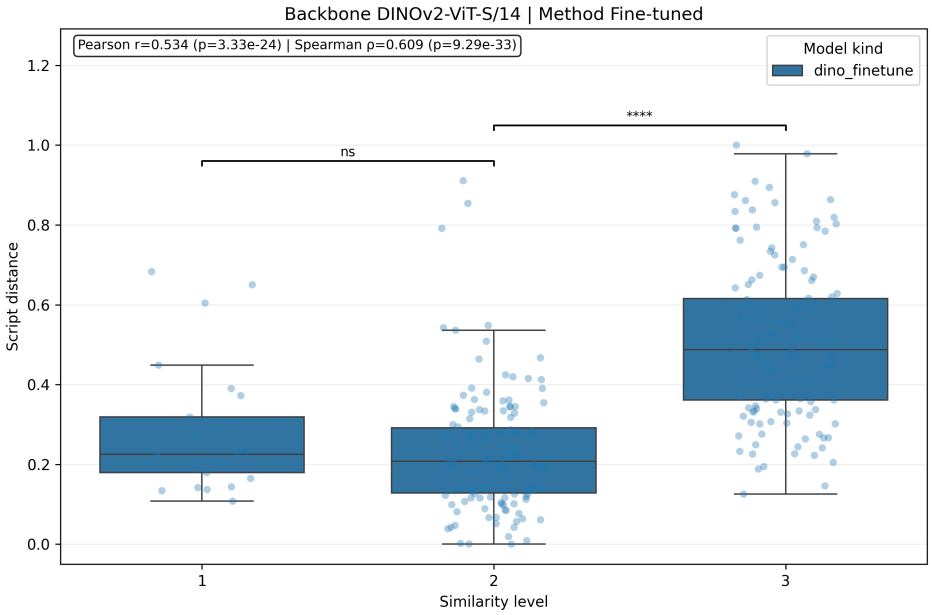


Figure 31: **Backbone: DINOv2-ViT-S/14 | Method: Fine-tuned.** Script-distance versus similarity-level analysis.