WHAT A CHOREOGRAPHY FINGERPRINT MUST FORGET: LEARNING MOTION TOKENS BEYOND RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Motion tokenizers trained to reconstruct human movement are widely used as general-purpose motion representations. We study whether their tokens identify a choreography when a different person performs it, and introduce the task of CHOREOGRAPHY FINGERPRINTING: given a short and possibly occluded fragment of a performance, a system must find other performances of the same choreography in an index that grows without retraining, and reject choreographies it does not contain. The task matters in practice: a viral choreography is repeated by thousands of people on social media, and courts have started to consider whether a sequence of movements was copied from another work. A fingerprint for this task has to ignore the performer, the camera and unseen parts while keeping the order of movements, and reconstruction requires neither. We show that codes can be invariant or have high entropy and still carry no information about the choreography. Following how dance notation records a choreography, as ordered movements of body parts with the floor path kept apart, we propose CHOREOMATCH: trained on choreography labels without a reconstruction target, it infers hidden body parts without treating them as evidence and stores each third of a second in a few bytes. To test identification beyond the studio, we introduce the CHOREOGRAPHY FINGERPRINTING DATASET of skeleton sequences extracted from amateur videos of viral choreographies filmed on phones, annotated by choreography. Our framework identifies choreographies in these videos and outperforms reconstruction tokenizers, suggesting that a choreography is recognized by what a representation is trained to keep rather than by what it can reconstruct.

1 INTRODUCTION

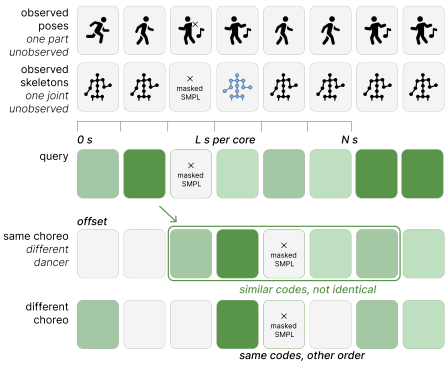

Figure 1: Choreography fingerprinting.

Fingerprinting is the task of identifying content from a compact representation of an observation (Cano et al., 2005). A useful fingerprint recognizes the same content despite specified changes in its appearance or recording, yet still distinguishes it from other items (Burges et al., 2003; Koval et al., 2007). Its design therefore depends on which differences preserve identity (Pizzi et al., 2022; Araz et al., 2026). Audio offers a well-studied example. The music-identification system SHAZAM (Wang, 2003) matches acoustic landmarks and their relative times to recognize short, distorted excerpts. Learned audio fingerprints suppress recording distortions while retaining identifying information (Chang et al., 2021).

Choreography raises a further difficulty: a routine can survive changes in the performer and in the execution itself (Oveneke et al., 2012; Aristidou et al., 2018). In music, the same distinction separates recording identification from cover-song identification (Serrà et al., 2008; Xun et al., 2023). Choreographic reuse has also prompted legal disputes. In *Hanagami v. Epic Games* (Ninth Circuit, 2023), a choreographer alleged that a *Fortnite* emote copied part of his

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.

work. In reversing the dismissal, the Ninth Circuit emphasized the arrangement of movements, not just isolated poses. Leaving the legal assessment aside, such cases motivate finding corresponding sequences across performances. In this paper, we propose the CHOREOGRAPHY FINGERPRINTING problem as an alternative lens for understanding learned motion representations (Zhu et al., 2023). The task is to identify a choreography from a short motion fragment *across different performances* (Fig. 1). We also introduce the CHOREOGRAPHY FINGERPRINTING DATASET, which comprises skeletal sequences extracted from videos on multiple social media platforms, choreography annotations, and an evaluation protocol (Section 3.2).

Reconstruction-trained motion tokens offer a natural starting point (van den Oord et al., 2017; Gökay et al., 2025; Zhang et al., 2023b). A representation that retains enough information to recover a movement might also support recognizing it when it is performed again (Wang & Neff, 2015; Petrovich et al., 2023). Yet a reconstruction-trained tokenizer, searched through token histograms, underperforms direct comparisons of pose and motion (Section C). This discrepancy motivates a closer examination of how *reconstructing an observed performance* relates to *identifying the underlying choreography*. We first show that perfect invariance does not guarantee choreography identification. Specifically, constant codes minimize the hard-token disagreement between an input and its transformed version, yet they carry no information about choreography identity. We further prove that even high marginal code entropy does not guarantee identity information (Proposition 1).

Examining the training target reveals a more specific limitation. Applying a different orthogonal transformation to each frame leaves the per-frame inter-joint distances unchanged, even when the joint trajectories differ. A per-frame distance target therefore cannot tell such motions apart (Section A.2). Quantization (Sedmidubský et al., 2020; Jégou et al., 2011) introduces two distinct losses: *replacing descriptors with centroids*, and *comparing their codes by exact equality*. Controls with fixed descriptors, assignments, and temporal matching separate the two. Centroid-similarity lookup removes the second loss but does not recover the original descriptors (Section A.3).

A fingerprint must therefore suppress variation irrelevant to identity while retaining the motion that distinguishes choreographies. We build CHOREOMATCH around three requirements. First, it is *trained for identity*: a Transformer (Vaswani et al., 2017) with factorized spatiotemporal attention (Bertasius et al., 2021) learns from choreography labels without a reconstruction target. The encoder keeps ordered joint histories, while a separate stream encodes pelvis translation and turning as a correction to the body description. Second, it is *aware of what it did not see*: hidden body parts may query visible ones but never count as evidence. Third, its codes are *compared by geometry* rather than by index. New choreographies are indexed by appending their codes, with the encoder and codebooks held fixed.

2 PRELIMINARIES

Motion as skeletons. We describe a dance by the 3D skeleton of its performer rather than by pixels (Fig. 2). The SMPL body model (Loper et al., 2015) gives joint positions $\mathbf{p}_{t,j}$ on a 30 Hz grid, where $j = 0$ is the pelvis and $j = 1, \ldots, J$ are the other joints. A mask $M \in \{0, 1\}^{T \times J}$ over the T frames marks each joint as observed ($M_{t,j} = 1$) or unseen ($M_{t,j} = 0$). The mask thus distinguishes an unseen joint from a joint observed at the origin. The joints are grouped into K body parts $\mathcal{P}_1, \ldots, \mathcal{P}_K$.

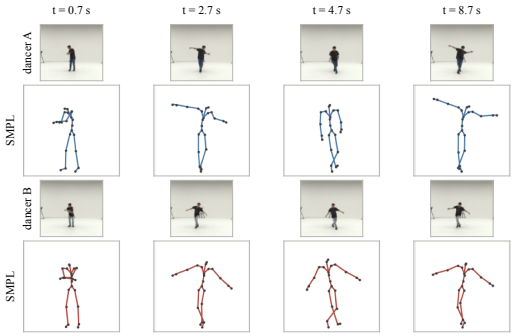

Figure 2: One AIST++ choreography danced by two dancers: video frames and their SMPL skeletons at the same moments.

Pelvis centering and root motion. The pose sequence $X = (\mathbf{x}_t)$ uses pelvis-relative joints, $\mathbf{x}_{t,j} = \mathbf{p}_{t,j} - \mathbf{p}_{t,0}$ for $j = 1, \ldots, J$. Centering removes *two different things*: where the dancer stands, which is a nuisance, and how the dancer travels, which is part of the choreography. We therefore keep the removed trajectory as root motion $\mathbf{r}_t = (v_{x,t}, v_{z,t}, \omega_t, a_t)$: planar velocity, turning rate and an availability bit. Velocities retain how the

dancer travels but discard where the dancer stands. Centering does *not* remove orientation, scale or proportions, so invariance to these must be learned.

Local descriptors and discrete codes. A performance is cut into cores of C frames, indexed by i. An encoder reads each core with w_c frames of context on each side and returns a descriptor $z_i \in \mathbb{R}^d$. A codebook $\mathcal{C} = \{c_n\}$ maps this descriptor to a code $k_i = \arg\min_n \|z_i - c_n\|$ and a reconstruction $\hat{z}_i = c_{k_i}$. Storing the code takes only a few bytes per core. A code names a region of descriptor space, not a choreography. Codes can thus be compared either by *equality of IDs* or by the *geometry* of their reconstructions. We write $[\cdot;\cdot]$ for concatenation and $\odot$ for the elementwise product.

3 CHOREOGRAPHY FINGERPRINTING

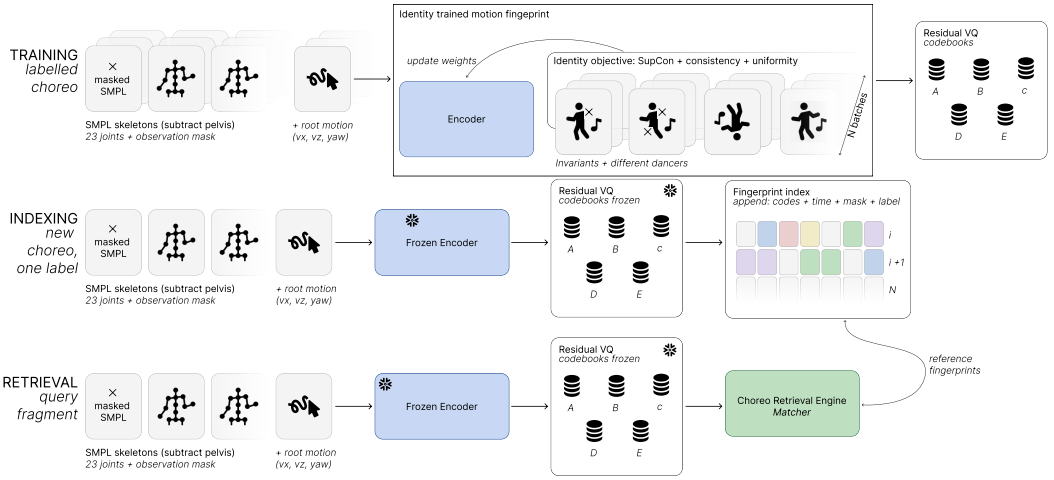

Figure 3: CHOREOMATCH. *Training* fits the encoder for identity and then the codebooks; *indexing* adds frozen fingerprints of new performances; *retrieval* matches a query against the index.

3.1 IDENTIFYING A CHOREOGRAPHY ACROSS PERFORMANCES

Audio fingerprinting recognizes one recording under distortion, whereas cover-song identification recognizes one composition across performers. Dance needs the second notion. A viral choreography is performed by thousands of people with different bodies, timing and cameras. What these performances share is the ordered arrangement of movements. We define CHOREOGRAPHY FINGERPRINTING as identifying a choreography from a short fragment of one of its performances. The task must ignore who dances, how the dance is filmed and what is hidden. Unlike action recognition or motion retrieval, it must also separate choreographies that share a style or moves (Fig. 5).

Formally, a performance X with mask M (Section 2) carries a choreography label $y(X)$ and a source recording $\sigma(X)$. A fingerprint ϕ maps the performance to a compact sequence of local signatures $(k_1, \ldots, k_N)$ drawn from an alphabet $\mathcal{A}$. Fingerprints are compared by a score $S(Q, X)$. For a query fragment Q and an index $\mathcal{I}$ of labelled references, the relevant set is

$$\mathcal{R}(Q) = \{X \in \mathcal{I} : \; y(X) = y(Q), \; \sigma(X) \neq \sigma(Q)\}, \tag{1}$$

that is, the references of the query's choreography that come from other recordings. The system ranks $\mathcal{I}$ by S and returns a prediction $\hat{y}(Q) \in \mathcal{Y} \cup \{\text{UNKNOWN}\}$, where $\mathcal{Y}$ is the label set. Answering UNKNOWN is correct exactly when $\mathcal{R}(Q) = \varnothing$. New choreographies enter $\mathcal{I}$ without retraining ϕ.

We instantiate this task in CHOREOMATCH, a framework of three operations (Fig. 3). *Training* first fits the encoder behind ϕ with an identity objective on labelled performances of different dancers. It then fits residual codebooks on the frozen descriptors. *Indexing* fingerprints a new performance with the encoder and codebooks both frozen, and appends the result to $\mathcal{I}$. *Retrieval* fingerprints a query in the same way, and the CHOREOGRAPHY RETRIEVAL ENGINE (CRE) scores it against every entry of $\mathcal{I}$. Section 4 describes each component.

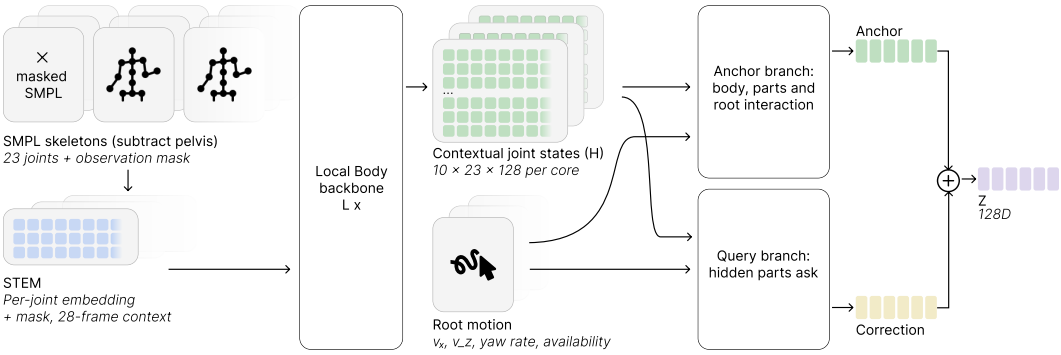

Figure 4: Encoder overview: a masked local backbone gives joint states H_i; the ordered anchor reads them into one descriptor, and the query branch adds a correction for hidden parts.

3.2 Corpora with different notions of sameness

Table 1: Corpora, grouped by what a shared label means.

Corpus	Performers	Poses	Same label
Independent performances			
AIST++	professionals	multi-view	routine
Ours, human	amateurs	monocular	trend
Repeated executions			
Ours, animations	avatars	monocular	emote render
AIST++ cross-cam.	professionals	per camera	execution
Action categories			
NTU RGB+D	actors	Kinect	action

We use five skeleton corpora (Table 1). AIST++ (Li et al., 2021), with professional dancers in a studio, is our development split. The human part of our Choreography Fingerprinting Dataset, with amateurs filmed on phones, is held out for zero-shot testing. The other three corpora serve as controls: their shared labels mark a repeated execution or only a common action (Shahroudy et al., 2016).

4 From Motion Tokens to Identity Fingerprints

4.1 Why a motion tokenizer is not a fingerprint

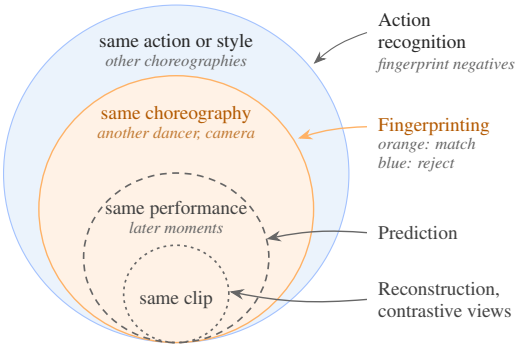

Figure 5: Choreography Fingerprinting must *match* other performances inside orange and *reject* the blue ring.

Which choreography does a short clip of a viral dance trend perform, whoever dances it? An obvious approach learns motion tokens by reconstructing the skeleton through a quantized bottleneck, and then compares their histograms. In a preliminary experiment (Section C), such tokens *describe the body, not the choreography* (Fig. 5): most codes are in use, yet retrieval stays weak. For our own encoder, an auxiliary pose decoder also established no gain (Section E.2). Moreover, for the same stored codes, requiring identical IDs retrieves far worse than comparing their reconstructions (Section E.1). A fingerprint should therefore be *trained for identity*, *aware of what it did not see* (Section 4.2), and *compared by geometry* (Section 4.3).

4.2 An observation-aware encoder

Intuition. Dance notation records a dance not as a set of poses but as a schedule of which body part moves, when, and in what order. Labanotation gives each body part a column along time (Fig. 6). Feuillet notation records the floor path as its own track and places the steps along it (Guest, 1970; Feuillet, 1700; Farnell, 2005). Such a score identifies one work across its performances (Goodman, 1968). Our encoder follows the same factorization (Fig. 4). It first *looks* at each joint in its moment

and then at the whole body. It then *writes* what it sees into an ordered score with limb and floor-path marks, and finally *reads again* for what the camera missed.

Looking: a masked local backbone. Each joint is first embedded from its positions and visibility in three consecutive frames, $h^0_{t,j} = M_{t,j}\left(g([\mathbf{x}_{t-1:t+1,j}; M_{t-1:t+1,j}]) + e_j\right)$, with a shared MLP g and a joint embedding e_j. Following factorized space–time attention for motion and video (Aksan et al., 2021; Bertasius et al., 2021), each of L blocks then updates

$$h \leftarrow M \odot \mathrm{FFN}\big(\mathrm{A}^M_{\mathrm{body}}(\mathrm{A}^M_{\mathrm{time}}(h))\big), \qquad (2)$$

where $\mathrm{A}^M_{\mathrm{time}}$ attends to the same joint within δ frames and $\mathrm{A}^M_{\mathrm{body}}$ to all joints of the same frame. The superscript M restricts keys to observed joints (Fig. 7; residuals and normalization are in Section D). A hidden joint therefore never serves as a key, and its state is reset to zero after every block. The states of core i form $H_i \in \mathbb{R}^{C\times J\times d}$.

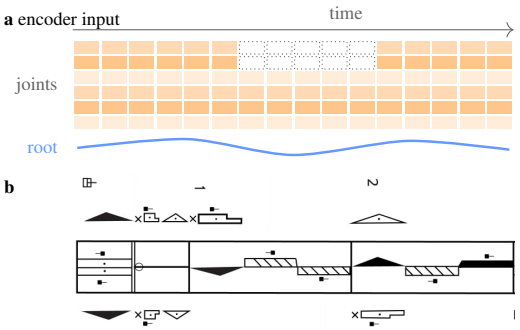

Figure 6: (a) Encoder input; (b) Labanotation staff, courtesy of the Dance Notation Bureau.

Writing the score: an ordered anchor. The anchor *concatenates rather than averages*, so it keeps the order of frames and joints (Fig. 8). Each joint's history over the core is projected to $U_{i,j}$. The joint vectors, in fixed order, are then projected to the body term (Eq. (3)). Coarser marks are added on top as corrections (Eq. (4)). The part term $\Delta^{\mathrm{parts}}_i$ comes from attention among the K part summaries, and the root term Δ^{root}_i from the ordered root motion of the core. In the interaction term Δ^{int}_i, root motion attends to the parts.

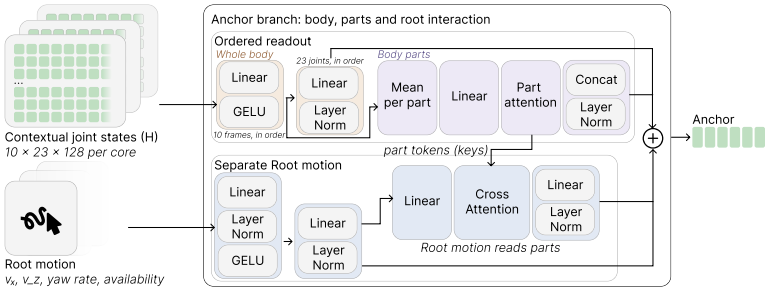

Figure 8: Ordered anchor (Eqs. (3) and (4)): histories and joints are concatenated in order; corrections are added on top.

$$U_{i,j} = \mathrm{GELU}\big(W_T[H_{i,1,j}; \ldots; H_{i,C,j}]\big), \qquad z^{\mathrm{body}}_i = \mathrm{LN}\big(W_J[U_{i,1}; \ldots; U_{i,J}]\big), \qquad (3)$$

$$z^{\mathrm{anc}}_i = z^{\mathrm{body}}_i + \alpha\Delta^{\mathrm{parts}}_i + \beta q_i \Delta^{\mathrm{root}}_i + \gamma q_i I_i \Delta^{\mathrm{int}}_i, \qquad (4)$$

where α, β, γ are learned scalar gates, $q_i \in [0, 1]$ is the fraction of core frames with an available root observation, and $I_i \in \{0, 1\}$ equals one only when the core has body, root and part support. Hidden parts are excluded from the part attention, and both root terms vanish when no root is observed ($q_i = 0$). Hence *no correction is computed from evidence that was not seen*. The hierarchy *follows dance notation*. The staff records the body, while body-region columns and floor plans annotate it (Guest, 1970). One step sequence can also recur on different floor paths (Pierce, 2002). Root motion therefore enters only as a correction: *the path refines the work rather than defines it*. Alternative designs are compared in Section E.4.

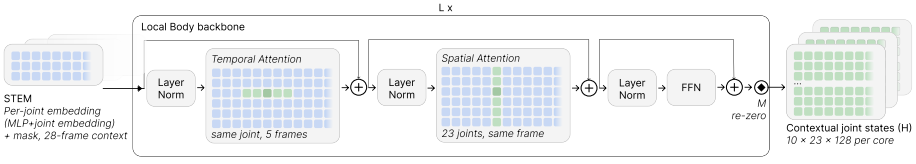

Figure 7: Local backbone (Eq. (2)): time, then body attention; hidden joints give no keys and are reset to zero after every block.

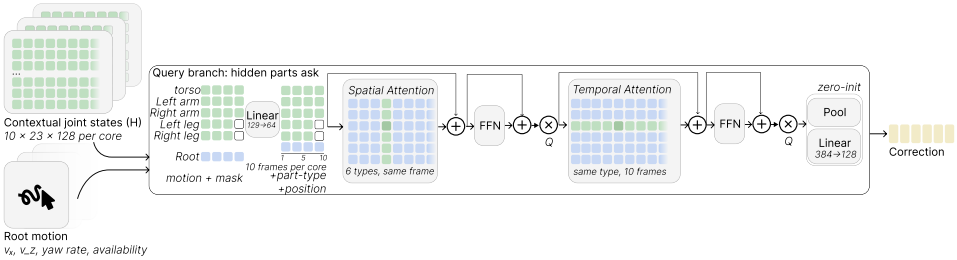

Figure 9: Query branch (Eqs. (5) and (6)). Hidden parts may query visible ones but never serve as evidence; a zero-initialized projection returns the correction.

Reading again: a query branch. A reconstruction model would draw a missing body part back in. For a fingerprint, this would *invent evidence*. The query branch therefore reads H_i again before the anchor compresses it (Fig. 9). For every frame t, it forms one token $V_{t,p}$ for each of the K parts and one token for root motion. A part token is built from the mean observed state of the part and its observed fraction. Two masks *separate permission from evidence*. The permission mask $\Pi_{t,p}$ is 1 if the token may hold a state, and the evidence mask $\mathcal{E}_{t,p}$ is 1 only if the part was observed. A hidden part thus has $\Pi = 1$ and $\mathcal{E} = 0$, so it may query other tokens but never serves as a key. A mixing step is

$$V' = V + \mathrm{A}^{\mathcal{E}}(\mathrm{LN}\, V), \qquad V \leftarrow \Pi \odot \big(V' + \mathrm{FFN}(\mathrm{LN}\, V')\big), \tag{5}$$

where $\mathrm{A}^{\mathcal{E}}$ is attention with keys restricted to $\mathcal{E} = 1$. One such step mixes the parts within a frame, and a second mixes the frames of each part. Pooling each token over the core and projecting the result gives the correction

$$z_i = z_i^{\mathrm{anc}} + \Delta_i^{\mathrm{query}}, \qquad \Delta_i^{\mathrm{query}} = W_0\,[\bar{V}_{i,1}; \ldots; \bar{V}_{i,K}; \bar{V}_{i,\mathrm{root}}], \quad W_0 = 0 \text{ at initialization.} \tag{6}$$

Inferred states never become keys, and the branch starts exactly at the anchor. In an unpaired comparison with the anchor alone, the branch coincided with higher retrieval under a hidden body part and similar clean retrieval (Section E.5).

Learning choreography identity, not poses. The encoder is trained *only to tell choreographies apart* (Algorithm 1). Training has two stages with the same objective: the anchor first, then the query branch with the anchor frozen. Each batch pairs two performances of each of several choreographies. Each clip is further occluded by a mask from an occlusion prior, which hides a part or frames. Let u_b be the normalized mean descriptor of clip b over its observed cores, and $P(b)$ the other clips of its choreography. The objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{SupCon}} + \lambda_c \mathcal{L}_{\mathrm{cons}} + \lambda_u \mathcal{L}_{\mathrm{uni}}, \tag{7}$$

Algorithm 1 Staged identity training

Require: $\mathcal{D} = \{(X_n, M_n, \mathbf{r}_n, y_n)\}$; parameters $\theta = (\theta_{\mathrm{anc}}, \theta_{\mathrm{q}})$; $\tau, \lambda_c, \lambda_u, \eta$; occlusion prior $\mathcal{O}$; transform prior $\mathcal{G}$

for $\vartheta \in (\theta_{\mathrm{anc}}, \theta_{\mathrm{q}})$ **do** ▷ anchor first, then query branch with the anchor frozen
  for $n_{\mathrm{step}} = 1, \ldots, N_{\mathrm{step}}$ **do**
    $\mathcal{B} \leftarrow$ two performances b of each choreography in a sampled set; $\rho \sim \mathcal{G}$ ▷ rotation and scale
    for $b \in \mathcal{B}$ **do**
      $\tilde{M}_b \leftarrow M_b \odot m_b, \quad m_b \sim \mathcal{O}$ ▷ hide a part or frames
      $(z_{b,i})_i \leftarrow f_\theta(X_b, \tilde{M}_b, \mathbf{r}_b)$; $(z'_{b,i})_i \leftarrow f_\theta(\rho X_b, \tilde{M}_b, \rho\, \mathbf{r}_b)$
      $u_b \leftarrow \sum_i \kappa_{b,i} z_{b,i} \,/\, \big\|\sum_i \kappa_{b,i} z_{b,i}\big\|$ ▷ $\kappa_{b,i} = 1$ if core i is observed
    end for
    $P(b) \leftarrow \{a \in \mathcal{B} \setminus \{b\} : y_a = y_b\}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{SupCon}}(u; P, \tau) + \lambda_c \mathcal{L}_{\mathrm{cons}}(z, z') + \lambda_u \mathcal{L}_{\mathrm{uni}}(u)$ ▷ Eq. (7)
    $\vartheta \leftarrow \vartheta - \eta\, \nabla_\vartheta \mathcal{L}$
  end for
end for
freeze θ; $(\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(R)}) \leftarrow \textsc{FitRVQ}\big(\{z_{n,i}\}\big)$ ▷ Eq. (8)
return $f_\theta, \mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(R)}$

where $\mathcal{L}_{\text{SupCon}}$ is supervised contrast (Khosla et al., 2020) over u with positives $P(b)$ and temperature τ. The consistency term $\mathcal{L}_{\text{cons}}$ is $1 - \cos(z_{b,i}, z'_{b,i})$ between a core and its rotated, rescaled copy, and $\mathcal{L}_{\text{uni}}$ is the uniformity loss (Wang & Isola, 2020) on u. Full forms are in Section D. Supervised contrast groups performances of one choreography, consistency keeps local descriptors stable under viewpoint and body scale, and uniformity prevents collapse. We use *no pose decoder*, because an auxiliary decoder established no consistent retrieval gain across observation conditions (Section E.2). Codebooks are also *fitted after training, not learned jointly*. Neither updating the encoder and codebooks together nor refining the fitted centroids afterwards established a gain (Section E.3).

4.3 The Choreography Retrieval Engine

With the encoder frozen, we fit residual quantization (Chen et al., 2010) with R codebooks $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(R)}$ on training descriptors. Each stage quantizes the residual left by the previous stages,

$$a_i^{(r)} = \arg\min_a \Big\| z_i - \sum_{l<r} \mathcal{C}^{(l)}_{a_i^{(l)}} - \mathcal{C}^{(r)}_a \Big\|, \qquad k_i = \big(a_i^{(1)}, \ldots, a_i^{(R)}\big), \qquad \hat{z}_i = \sum_{r=1}^{R} \mathcal{C}^{(r)}_{a_i^{(r)}}, \quad (8)$$

so every core is stored in a few bytes.

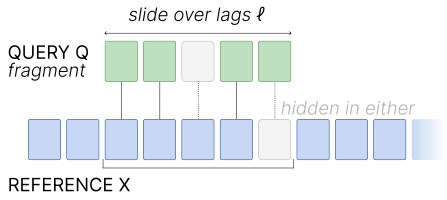

Figure 10: The CRE. Each box is the code of one core; the query *slides* over the reference, cores observed in both *vote* (solid), hidden ones *abstain* (dotted), and the bracket marks the best lag ℓ^*.

The CRE scores a query Q against every reference X in the index (Fig. 10). It has four properties. First, *codes are compared by geometry*, as in product-quantization search (Jégou et al., 2011). Two codes k, k' with reconstructions $\hat{z}, \hat{z}'$ have similarity $s(k, k') = \cos(\hat{z}, \hat{z}')$, which compares reconstructions rather than the equality of IDs. Nearby moments that fall into neighbouring cells therefore still match (Table 3). Second, *only observed evidence votes*. The query is scored at its best time lag,

$$S(Q, X) = \max_{\ell \in \Lambda} \frac{1}{|\Omega_\ell|} \sum_{i \in \Omega_\ell} s\big(k_i^Q, k_{i+\ell}^X\big), \quad (9)$$

where Ω_ℓ holds the cores *sufficiently observed in both* sequences. Hidden cores keep their time position but never vote. Third, *the match is localized*: the maximizing lag ℓ^* places the fragment inside the reference. Fourth, *unindexed choreographies are rejected* by a calibrated threshold on S (Section 4.4). Elastic alignment of the same codes did not improve retrieval (Section E.6). Dynamic time warping (Rakthanmanon et al., 2012) applied to skeletons remains strong on short, fully observed fragments (Section 5).

4.4 Indexing and Rejection

A new performance is encoded with the frozen encoder and codebooks. Its codes are appended to $\mathcal{I}$ together with their timestamps and mask, *without retraining*. For a query Q, let $X^* = \arg\max_{X \in \mathcal{I},\, \sigma(X) \neq \sigma(Q)} S(Q, X)$ be the best-scoring reference from another recording. The engine instantiates the decision of Section 3 as

$$\hat{y}(Q) = \begin{cases} y(X^*), & S(Q, X^*) \geq \xi, \\ \text{UNKNOWN}, & \text{otherwise.} \end{cases} \quad (10)$$

It also returns the ranking of $\mathcal{I}$ and the lag ℓ^*. The threshold ξ is *calibrated on held-out data* for a target false-acceptance rate and then kept fixed.

5 Experiments

5.1 Experimental setup

We evaluate on the corpora of Section 3.2. On AIST++, each query is observed clean, with a body part hidden, or with periodic gaps, and Table 2 reports the mean over these three conditions. Ranking is scored by mAP over the full relevant set $\mathcal{R}(Q)$ (Eq. (1)) and by MRR. Rejection is scored by correct (CA) and false (FA) acceptance at a threshold calibrated for 5% FA. We compare against four baseline families: raw-pose statistics and alignment, pose codes, pretrained motion encoders (Zhu et al., 2023; Petrovich et al., 2023), and reconstruction motion tokens (details in Section B).

Table 2: Choreography retrieval on AIST++ (Li et al., 2021) and our CHOREOGRAPHY FINGERPRINTING DATASET (ours). AIST++ reports mAP on whole recordings; CHOREOGRAPHY FINGERPRINTING DATASET reports mAP and MRR on human performances of viral trends.

Methods	AIST++ mAP	CHOREOGRAPHY FINGERPRINTING DATASET mAP	MRR
Baseline			
Chance level (random ranking)	13.5	3.0	–
Pose statistics and alignment			
DTW, joint trajectories	72.6	82.6	86.1
DTW, banded	69.5	82.7	86.4
Skeleton DTW, masked	80.3	–	–
Pose codes			
PQ words over raw pose (8×256)	63.3	88.2	92.1
DPQ, retrieval-trained codebooks (8×256)	76.5	83.9	84.7
Two-stage: PQ shortlist + DTW	68.6	82.7	86.3
Two-stage: DPQ shortlist + DTW	75.8	–	–
Two-stage: ordered shortlist + DTW	–	82.8	–
Learned motion representations			
MotionBERT, mean-pooled (Zhu et al., 2023)	19.1	74.2	81.4
PQ words over MotionBERT	17.6	–	–
TMR motion encoder (Petrovich et al., 2023)	52.8	82.7	86.3
TMR, window mean	54.2	82.9	88.5
PQ words over TMR latents	54.6	84.1	90.2
Reconstruction motion tokens			
Reconstruction tokenizer, continuous	14.4	77.6	79.0
Reconstruction tokenizer, codes	16.2	66.4	–
+ identity supervision	38.3	82.9	–
MoMask, code histograms (Guo et al., 2024)	16.0	79.8	–
MoMask, mean embeddings	10.8	82.4	–
MoMask, DTW over embeddings	13.8	87.2	–
Ours			
CHOREOMATCH, continuous descriptors	95.1	92.7	96.2
CHOREOMATCH, codes (CRE)	89.8	91.7	94.6

5.2 MAIN RESULTS

CHOREOMATCH *outperforms every baseline family on both corpora* (Table 2). On AIST++, it reaches 95.1 mAP with continuous descriptors and 89.8 with codes. The strongest alignment baseline reaches 80.3, and pretrained encoders reach at most 54.6. On CHOREOGRAPHY FINGERPRINTING DATASET, which CHOREOMATCH never saw during training, it reaches 92.7 and 91.7. The strongest baseline there is pose product quantization, at 88.2. Reconstruction motion tokens, including the public MoMask tokenizer, stay close to chance on AIST++ (at most 16.2 against 13.5), and identity supervision raises them only to 38.3. Their codes are also the weakest entry on CHOREOGRAPHY FINGERPRINTING DATASET, as Section 4.1 predicts. Compressing to a few bytes per core costs 5.3 points on AIST++ and only 1.0 on CHOREOGRAPHY FINGERPRINTING DATASET.

5.3 WHAT MAKES A FINGERPRINT

Each requirement of Section 4.1 is necessary, and reconstruction substitutes for none of them (Table 3). First, *what turns motion tokens into a fingerprint is the training signal, not the codebook*. Training the encoder for identity rather than reconstruction moves retrieval from near chance to the top of Table 2. Reintroducing reconstruction gives nothing back, whether through an auxiliary decoder, jointly learned codebooks, or codebooks refined afterwards. Second, *knowing what was not observed* helps keep the fingerprint stable when only part of the body is visible: with a body part hidden, the query branch adds 7.9 points while clean

Table 3: What makes a fingerprint (AIST++, mAP %); one block per requirement of Section 4.1.

Comparison	AIST++ mAP from	to	AIST++ Difference Δ [95% CI]
Trained for identity			
Reconstruction tokens → CHOREOMATCH codes	16.2	89.8	$+73.6$
+ auxiliary pose decoder	89.9	89.4	CI $\ni 0$
Encoder and codebooks trained jointly	–	–	-0.7 [-2.7, 0.5]
Codebook refinement after fitting, part hidden	88.1	87.9	-0.2, CI $\ni 0$
Aware of what it did not see			
+ query branch, part hidden†	74.3	82.2	$+7.9$
+ query branch, clean†	90.0	89.9	-0.1
Compared by geometry			
ID equality → reconstruction cosine, part hidden	38.0	88.1	$+50.1$
Single codebook → residual, part hidden	83.7	88.1	$+4.4$ [2.3, 7.0]

Rows compare matched runs trained with identical settings unless marked; the final model is in Table 2.
† Unpaired: one anchor against the mean of three branch seeds.

retrieval is unchanged. Third, *discrete codes are only as good as their comparison*: ID matching discards most of what geometry retains.

5.4 Occlusion and rejection

Robustness and rejection are where a learned fingerprint departs most from alignment on raw poses (Fig. 11). Alignment inherits every gap in the skeleton. The fingerprint instead describes what remains visible, so its ranking *degrades gracefully as observations disappear*. With one arm hidden, skeleton DTW drops from 86.0 to 69.7 mAP, whereas ChoreoMatch loses only 2.2 points with continuous descriptors and 4.5 with codes. Its scores are also calibrated well enough to *answer "unknown"*. At the same false-acceptance target, it accepts far more correct answers. Elastic alignment still keeps an edge on short, fully observed fragments. This suggests combining elastic alignment with learned codes.

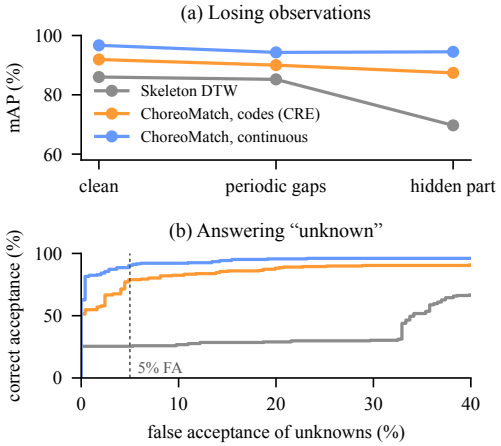

Figure 11: AIST++ whole recordings. (a) mAP as observations are lost; (b) Correct vs. false acceptance, pooled over conditions.

6 Related Work

Identifying a work across versions. Audio fingerprints recognise one recording under distortion (Wang, 2003; Chang et al., 2021), as copy detection matches a shared source (Pizzi et al., 2022; He et al., 2022); cover-song identification instead matches new performances of a composition (Serrà et al., 2008; Xun et al., 2023), an idea recently revisited in the context of short queries (Du et al., 2023; Serrà et al., 2025). Choreography Fingerprinting is its analogue for dance, where a version also differs in body and camera, and parts of the performer may be out of view.

Motion and choreography retrieval. A long line of work has explored motion retrieval with body-part signatures, motifs or segment vocabularies under elastic alignment (Wang & Neff, 2015; Aristidou et al., 2018; Sedmidubský et al., 2020; 2023; Jiang et al., 2023; Sakoe & Chiba, 1978; Rakthanmanon et al., 2012). Closest to our work is choreography retrieval from K-pop recordings (Kim et al., 2015), alongside music retrieval by dancing (Tsuchida et al., 2019); these measure similarity in a curated database but neither separate choreographies that share moves nor reject queries that match nothing. Skeleton and text–motion representations are trained for action categories or descriptions (Zhang et al., 2023a; Do et al., 2026; Peng et al., 2023; Petrovich et al., 2023; Li et al., 2025; Messina et al., 2025; Zhang et al., 2026), which many choreographies share.

Motion tokenizers and compact codes. Motion tokenizers and encoders learn to reconstruct or generate poses (Zhu et al., 2023; van den Oord et al., 2017; Zhang et al., 2023b; Guo et al., 2024; Zou et al., 2024; Gökay et al., 2025; Huang et al., 2026), often with factorized space–time attention (Aksan et al., 2021; Bertasius et al., 2021). Their objective rewards recovering who performs a movement and how, which a fingerprint must forget, and does not require identifying the choreography. Vector search instead compares codes through their reconstructions (Jégou et al., 2011; Chen et al., 2010; Babenko & Lempitsky, 2014), an idea recently revisited with neural codebooks (Huijben et al., 2024; Vallaeys et al., 2025). We also compare ChoreoMatch with various methods based on reconstruction tokens, pretrained encoders and product quantization.

7 Conclusion

We asked whether reconstruction-trained motion tokens recognize a choreography danced by someone else. They do not: reconstruction *describes the body, not the choreography*. ChoreoMatch is instead *trained for identity*, *aware of what it did not see* and *compared by geometry*. It indexes new performances without retraining, rejects unindexed choreographies, and transfers from studio dancers to amateurs filmed on phones. Relying only on skeletons, it misses fingers, props and music, and inherits pose errors. It also assumes one performer per clip. Learning fingerprints from video and identifying group choreographies remain open.

Reproducibility Statement

Code implementing ChoreoMatch, the baselines and the evaluation protocol is available in an anonymous repository (https://anonymous.4open.science/r/choreo_match-anonymous). The task, relevance set and metrics are defined in Sections 3 and 5.1; the encoder, the training procedure (Algorithm 1) and the Choreography Retrieval Engine (CRE) in Section 4, with architecture details and hyperparameters in Section D; The Choreography Fingerprinting Dataset (pose sequences and choreography annotations) and trained checkpoints will be released on Hugging Face upon publication.

Ethics Statement

The Choreography Fingerprinting Dataset is built from publicly posted social-media videos of dance trends. We convert each video into a skeleton sequence and keep only joint positions and choreography labels: we do not release videos, audio, faces, usernames or account links, raw videos were deleted after annotation. Skeletons remove appearance and identity cues, although motion can still carry traits of an individual performer; we release the data for research use only and will remove any sequence on request. ChoreoMatch matches sequences of movement and makes no judgment about authorship or copying; it is not intended as a tool for legal decisions such as *Hanagami v. Epic Games* (Ninth Circuit, 2023).

AI Use Statement

We used several general-purpose large language models to edit the text. When building the Choreography Fingerprinting Dataset, an LLM first labelled scraped videos as containing a dance or not; two human annotators then reviewed the videos retained by this filter.

References

Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3D human motion prediction. In *International Conference on 3D Vision (3DV)*, pp. 565–574, 2021. doi: 10.1109/3DV53792.2021.00066.

R. Oguz Araz, Joan Serrà, Yuki Mitsufuji, Xavier Serra, and Dmitry Bogdanov. Unified music identification for tracks and versions, 2026. URL https://arxiv.org/abs/2608.19919. Accepted to ISMIR 2026.

Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. Deep motifs and motion signatures. *ACM Transactions on Graphics*, 37(6):187:1–187:13, 2018. doi: 10.1145/3272127.3275038. URL https://andreasaristidou.com/DeepSignatures.

Artem Babenko and Victor Lempitsky. Additive quantization for extreme vector compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 931–938, 2014. doi: 10.1109/CVPR.2014.124. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Babenko_Additive_Quantization_for_2014_CVPR_paper.html.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021. URL https://proceedings.mlr.press/v139/bertasius21a.html.

Christopher J. C. Burges, John C. Platt, and Soumya Jana. Distortion discriminant analysis for audio fingerprinting. *IEEE Transactions on Speech and Audio Processing*, 11(3):165–174, 2003. doi: 10.1109/TSA.2003.811538. URL https://doi.org/10.1109/TSA.2003.811538.

Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 41(3):271–284, 2005. doi: 10.1007/s11265-005-4151-3. URL https://link.springer.com/article/10.1007/s11265-005-4151-3.

Sungkyun Chang, Donmoon Lee, Jeongsoo Park, Hyungui Lim, Kyogu Lee, Karam Ko, and Yoonchang Han. Neural audio fingerprint for high-specific audio retrieval based on contrastive learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3025–3029, 2021. doi: 10.1109/ICASSP39728.2021.9414337. URL https://arxiv.org/abs/2010.11910.

Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010. doi: 10.3390/s101211259. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3231071/.

Jeonghyeok Do, Yun Chen, Geunhyuk Youk, and Munchurl Kim. Less is more: Compact-token masked feature prediction for skeleton representation learning. arXiv preprint arXiv:2603.10648v3, 2026. URL https://arxiv.org/abs/2603.10648v3. Version 3, 3 August 2026.

Xingjian Du, Zijie Wang, Xia Liang, Huidong Liang, Bilei Zhu, and Zejun Ma. ByteCover3: Accurate cover song identification on short queries. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095389. URL https://arxiv.org/abs/2303.11692.

Brenda Farnell. Movement notation systems. *Journal for the Anthropological Study of Human Movement*, 13(3):145–170, 2005. URL https://jashm.press.uillinois.edu/13.3/13-3Movement_Farnell145-170.pdf.

Raoul-Auger Feuillet. *Chorégraphie, ou l'art de décrire la dance, par caractères, figures et signes démonstratifs*. Chez l'auteur et Michel Brunet, Paris, 1700.

Uzay Gökay, Federico Spurio, Dominik R. Bach, and Juergen Gall. Skeleton motion words for unsupervised skeleton-based temporal action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12101–12111, October 2025. doi: 10.1109/ICCV51701.2025.01125. URL https://openaccess.thecvf.com/content/ICCV2025/html/Gokay_Skeleton_Motion_Words_for_Unsupervised_Skeleton-Based_Temporal_Action_Segmentation_ICCV_2025_paper.html.

Nelson Goodman. *Languages of Art: An Approach to a Theory of Symbols*. Bobbs-Merrill, Indianapolis, 1968.

Ann Hutchinson Guest. *Labanotation, or Kinetography Laban: The System of Analyzing and Recording Movement*. Theatre Arts Books, New York, 1970.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024. doi: 10.1109/CVPR52733.2024.00186. URL https://arxiv.org/abs/2312.00063.

Sifeng He, Xudong Yang, Chen Jiang, Gang Liang, Wei Zhang, Tan Pan, Qing Wang, Furong Xu, Chunguang Li, JinXiong Liu, Hui Xu, Kaiming Huang, Yuan Cheng, Feng Qian, Xiaobo Zhang, and Lei Yang. A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21054–21063, 2022. doi: 10.1109/CVPR52688.2022.02041. URL https://openaccess.thecvf.com/content/CVPR2022/html/He_A_Large-Scale_Comprehensive_Dataset_and_Copy-Overlap_Aware_Evaluation_Protocol_for_CVPR_2022_paper.html.

Tianlv Huang, Hetian Guo, Ziyi Cai, Song Wang, Yanping Zhang, Zipei Fan, Xuan Song, Guangming Wu, and Xin Zheng. SeMoCo: A semantic-first motion codec for motion language modeling, 2026. URL https://arxiv.org/abs/2608.24334v2. Version 2.

Iris A. M. Huijben, Matthijs Douze, Matthew J. Muckley, Ruud Van Sloun, and Jakob Verbeek. Residual quantization with implicit neural codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20682–20699. PMLR, 2024. URL https://proceedings.mlr.press/v235/huijben24a.html.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57. URL https://doi.org/10.1109/TPAMI.2010.57.

Zi-Fei Jiang, Wei Li, Yan Huang, Yi-Long Yin, C.-C. Jay Kuo, and Jing-Liang Peng. PESTA: An elastic motion capture data retrieval method. *Journal of Computer Science and Technology*, 38(4): 867–884, 2023. doi: 10.1007/s11390-023-3140-y. URL https://jcst.ict.ac.cn/cn/article/pdf/preview/10.1007/s11390-023-3140-y.pdf.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dohyung Kim, Minsu Jang, Youngwoo Yoon, and Jaehong Kim. Choreography retrieval from the Korean POP dance motion capture database with low-cost depth cameras. In *Advances in Computer Science and Ubiquitous Computing*, volume 373 of *Lecture Notes in Electrical Engineering*, pp. 805–811. Springer, Singapore, 2015. doi: 10.1007/978-981-10-0281-6_113. URL https://link.springer.com/chapter/10.1007/978-981-10-0281-6_113.

Oleksiy Koval, Sviatoslav Voloshynovskiy, Fokko Beekhof, and Thierry Pun. Decision-theoretic consideration of robust perceptual hashing: link to practical algorithms. In *WaCha2007, Third WAVILA Challenge*, Saint Malo, France, June 2007. URL https://archive-ouverte.unige.ch/unige:47800.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI Choreographer: Music conditioned 3D dance generation with AIST++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13381–13392, 2021. doi: 10.1109/ICCV48922.2021.01315. URL https://arxiv.org/abs/2101.08779.

Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence T. Yang. LaMP: Language-motion pretraining for motion generation, retrieval, and captioning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d1c88f9790765146ec8fb5d02e5653a0-Paper-Conference.pdf.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), November 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

Nicola Messina, Jan Sedmidubský, Fabrizio Falchi, and Tomáš Rebok. Joint-dataset learning and cross-consistent regularization for text-to-motion retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2025. doi: 10.1145/3744565. URL https://doi.org/10.1145/3744565.

Ninth Circuit. Hanagami v. Epic Games, Inc. United States Court of Appeals, Opinion No. 22-55890, November 2023. URL https://cdn.ca9.uscourts.gov/datastore/opinions/2023/11/01/22-55890.pdf. Filed November 1, 2023.

Meshia Cédric Oveneke, Valentin Enescu, and Hichem Sahli. Real-time dance pattern recognition invariant to anthropometric and temporal differences. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 407–419, 2012. doi: 10.1007/978-3-642-33140-4_36.

Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen. Delving deep into one-shot skeleton-based action recognition with diverse occlusions. *IEEE Transactions on Multimedia*, 25:1489–1504, 2023. doi: 10.1109/TMM.2023.3235300. URL https://arxiv.org/abs/2202.11423.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *ICCV*, pp. 9454–9463, 2023. doi: 10.1109/ICCV51070.2023.00870.

Ken Pierce. Choreographic structure in dances by Feuillet. In *Proceedings of the Society of Dance History Scholars, Twenty-Fifth Annual Conference*, pp. 96–106, Philadelphia, 2002.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14512–14522, 2022. doi: 10.1109/CVPR52688.2022.01413. URL https://openaccess.thecvf.com/content/CVPR2022/html/Pizzi_A_Self-Supervised_Descriptor_for_Image_Copy_Detection_CVPR_2022_paper.html.

Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD*, pp. 262–270, 2012. doi: 10.1145/2339530.2339576.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055.

Jan Sedmidubský, Petra Budíková, Vlastislav Dohnal, and Pavel Zezula. Motion words: A text-like representation of 3D skeleton sequences. In *Advances in Information Retrieval (ECIR)*, pp. 527–541. Springer, 2020. doi: 10.1007/978-3-030-45439-5_35. URL https://www.fi.muni.cz/~xkohout7/Research/vlastni/ConferencePapers/ECIR20/final.pdf.

Jan Sedmidubský, Fabio Carrara, and Giuseppe Amato. SegmentCodeList: Unsupervised representation learning for human skeleton data retrieval. In *Advances in Information Retrieval*, volume 13981 of *Lecture Notes in Computer Science*, pp. 110–124. Springer, 2023. doi: 10.1007/978-3-031-28238-6_8. URL https://link.springer.com/chapter/10.1007/978-3-031-28238-6_8.

Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008. doi: 10.1109/TASL.2008.924595. URL https://ieeexplore.ieee.org/document/4523006.

Joan Serrà, R. Oguz Araz, Dmitry Bogdanov, and Yuki Mitsufuji. Supervised contrastive learning from weakly-labeled audio segments for musical version matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53923–53939. PMLR, 2025. URL https://proceedings.mlr.press/v267/serra25a.html.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010–1019, 2016. doi: 10.1109/CVPR.2016.115.

Shuhei Tsuchida, Satoru Fukayama, and Masataka Goto. Query-by-dancing: A dance music retrieval system based on body-motion similarity. In *MultiMedia Modeling (MMM 2019)*, volume 11295 of *Lecture Notes in Computer Science*, pp. 251–263. Springer Science+Business Media, 2019. doi: 10.1007/978-3-030-05710-7_21. URL https://doi.org/10.1007/978-3-030-05710-7_21.

Théophane Vallaeys, Matthew Muckley, Jakob Verbeek, and Matthijs Douze. QINCo2: Vector compression and search with improved implicit neural codebooks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/pdf?id=2zMHHZ569S.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Avery Wang. An industrial-strength audio search algorithm. In *ISMIR*, 2003. doi: 10.5281/zenodo.1416339.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

Yingying Wang and Michael Neff. Deep signatures for indexing and retrieval in large motion databases. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pp. 37–45, 2015. doi: 10.1145/2822013.2822024. URL https://www.cs.ucdavis.edu/~neff/papers/MIG_2015_DeepSignature.pdf.

Jiahao Xun, Shengyu Zhang, Yanting Yang, Jieming Zhu, Liqun Deng, Zhou Zhao, Zhenhua Dong, Ruiqi Li, Lichao Zhang, and Fei Wu. DisCover: Disentangled music representation learning for cover song identification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 453–463, 2023. doi: 10.1145/3539618.3591664. URL https://arxiv.org/abs/2307.09775.

Jiahang Zhang, Lilang Lin, and Jiaying Liu. Prompted contrast with masked motion modeling: Towards versatile 3D action representation learning. In *Proceedings of the ACM International Conference on Multimedia*, pp. 7175–7183, 2023a. doi: 10.1145/3581783.3611774. URL https://arxiv.org/abs/2308.03975.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, pp. 14730–14740, 2023b. doi: 10.1109/CVPR52729.2023.01415.

Yao Zhang, Zhuchenyang Liu, Yanlan He, Thomas Ploetz, and Yu Xiao. Fine-grained motion retrieval via joint-angle motion images and token-patch late interaction. arXiv preprint arXiv:2603.09930v2, 2026. URL https://arxiv.org/abs/2603.09930v2. Version 2, 30 June 2026.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. MotionBERT: A unified perspective on learning human motion representations. In *ICCV*, pp. 15039–15053, 2023. doi: 10.1109/ICCV51070.2023.01385.

Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. ParCo: Part-coordinating text-to-motion synthesis. In *European Conference on Computer Vision*, pp. 126–143, 2024. doi: 10.1007/978-3-031-72992-8_8. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/07265.pdf.

A What a choreography fingerprint must forget

This section makes precise three observations of the introduction. Throughout, X is a core, y is the choreography it belongs to, a tokenizer ψ gives each core a code from a finite alphabet $\mathcal{A}$, and a nuisance change ν (another performer, camera or observation mask) leaves y unchanged.

A.1 Agreement and code usage do not imply identity

A tokenizer is often judged by two properties: its codes should agree when the same movement is seen differently, that is, the disagreement $D(\psi) = \Pr[\psi(X) \neq \psi(\nu X)]$ should be small, and all its codes should be used, that is, the entropy $H(\psi(X))$ should be high. Neither property asks the codes to say which choreography is danced.

Proposition 1. *A constant code has $D(\psi) = 0$ and carries no information about y. More generally, let a feature U be unchanged by every ν, independent of y, and divisible into $|\mathcal{A}|$ sets of equal probability. A code that depends only on U can then have $D(\psi) = 0$, use every code equally often, and still have $I(\psi(X); y) = 0$.*

Proof. Such a code gives X and νX the same value, so $D(\psi) = 0$. Mapping the $|\mathcal{A}|$ equal-probability sets of U to distinct codes makes every code equally frequent. Since ψ sees X only through U, $I(\psi(X); y) \leq I(U; y) = 0$. □

For example, the position of a core within its recording survives any change of performer or camera and is essentially unrelated to the choreography. Identity therefore has to be encouraged by the objective itself, as in Section 4.2.

A.2 A per-frame distance target ignores rotation

The distances $\|\mathbf{x}_{t,j} - \mathbf{x}_{t,j'}\|$ between the joints of frame t do not change when that frame is rotated or mirrored, and every frame may be rotated differently. A body turning on the spot and a body standing still therefore give the same target, as do a movement and its mirror image. A tokenizer trained to reproduce this target is not required to encode how the body turns over time, which choreographies use.

A.3 Comparing codes by geometry

Two codes can be compared by equality, $\mathbf{1}[k_i = k_j]$, or by the cosine of their reconstructions, $\cos(\hat{z}_i, \hat{z}_j)$ (Eq. (8)). Equal codes have equal reconstructions and thus cosine one, so geometry keeps everything equality sees; in addition, it ranks pairs whose codes differ, which equality treats as unrelated even when they fall into neighbouring cells. The gap to the descriptors themselves is bounded by the quantization error: since normalizing moves a vector by at most twice its relative error,

$$\big|\cos(\hat{z}_i, \hat{z}_j) - \cos(z_i, z_j)\big| \leq \frac{2\varepsilon_i}{\|z_i\|} + \frac{2\varepsilon_j}{\|z_j\|}, \qquad \varepsilon_i = \|z_i - \hat{z}_i\|,$$

and only a finer quantizer tightens this bound.

A.4 Chance level

If a query has $|\mathcal{R}|$ relevant references among the $|\mathcal{I}_Q|$ references ranked at random, where $\mathcal{I}_Q$ excludes its own recording, its expected average precision is

$$\mathbb{E}[\mathrm{AP}] = \frac{1}{|\mathcal{I}_Q|} \sum_{m=1}^{|\mathcal{I}_Q|} \frac{1}{m} \Big(1 + (m-1)\frac{|\mathcal{R}| - 1}{|\mathcal{I}_Q| - 1}\Big),$$

which gives the chance row of Table 2. It is well above $|\mathcal{R}|/|\mathcal{I}_Q|$ when $|\mathcal{R}|$ is small, because a single early hit already earns a high precision.

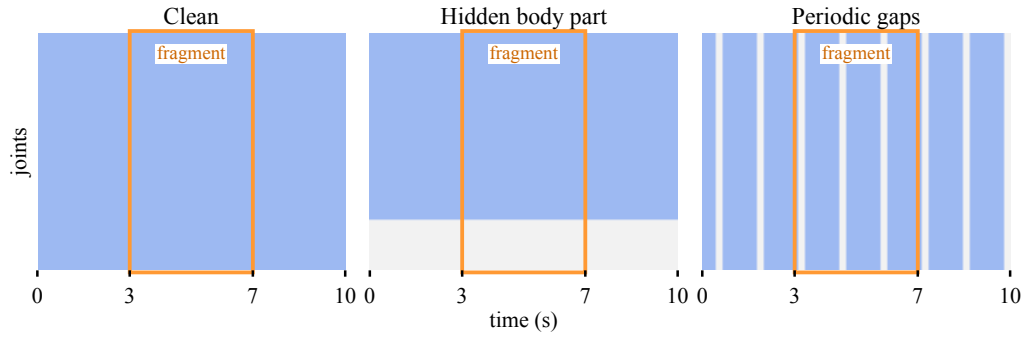

Figure 12: Observation conditions of a query: observed joints in blue, hidden ones in grey. A *hidden body part* removes one body part for the whole recording; *periodic gaps* remove the whole body for 8 of every 40 frames. A *fragment* is a 4 s window cut after masking.

B Evaluation protocol

B.1 Queries and observation masks

Every recording with at least one relevant reference (Eq. (1)) serves as a known query against the clean recordings of the same split; references from the query's own source are excluded. A *whole* query is the full recording. A *fragment* is a 4 s window (120 frames) whose start lies on a 1 s grid and is chosen by hashing the recording ID, independently of its label. Each query is evaluated under three observation conditions (Fig. 12): *clean*; *hidden body part*, in which one body part, five joints from collarbone to hand on one side, is hidden for the whole recording; and *periodic gaps*, in which the whole body is hidden for 8 of every 40 frames. The side of the body part and the phase of the gaps are also set by hashing the recording ID. Masks are placed on the full recording before a fragment is cut, so a fragment inherits the gaps at their original positions. For rejection, each query is also run *as unknown*: every reference of its choreography is removed from the index before search.

B.2 Metrics and uncertainty

Ranking. Average precision is computed over the full relevant set $\mathcal{R}(Q)$; relevant references that receive no valid score ($S = -\infty$) count as misses. mAP averages it over known queries, and MRR averages the reciprocal rank of the first relevant reference. Chance levels follow Section A.4.

Rejection. Correct acceptance (CA) is the fraction of known queries whose top reference is accepted and carries the correct label; false acceptance (FA) is the fraction of unknown queries that are accepted.

Uncertainty. Intervals are 95% percentile intervals from 2000 bootstrap resamples that draw recording sources and choreographies as clusters. Differences between two systems are *paired*: both are evaluated on the same resamples.

C PRELIMINARY EXPERIMENT: STT, VQ AND A TWO-STAGE ENGINE

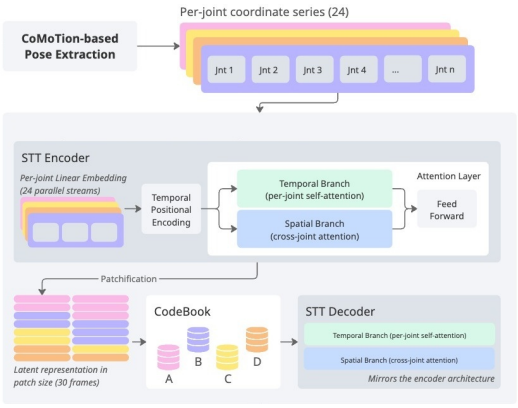

Figure 13: **Reconstruction-tokenizer baseline.** An STT encoder embeds each joint, latent patches are quantized into motion tokens with an EMA codebook, and an STT decoder reconstructs the motion.

This baseline follows the tokenizer route of Section 4.1 (Fig. 13).

Tokenizer. A spatio-temporal Transformer with parallel temporal and spatial attention encodes short windows of the skeleton, and a vector-quantized bottleneck (van den Oord et al., 2017) with 512 codes, updated by exponential moving averages, turns each window into a motion word $w_i \in \{1, \dots, 512\}$. The model is trained to reconstruct the skeleton from its words.

Retrieval engine. Search uses a two-stage design common in motion-word retrieval (Sedmidubský et al., 2020). The first stage represents a sequence by the normalized histogram of its words, compares histograms by cosine and keeps a shortlist of references. The second stage re-ranks the shortlist by a fixed weighted combination of the histogram cosine and similarities from edit distances and longest common subsequences between word sequences, all computed on word IDs with an exact-match cost.

Outcome. The tokenizer uses 429 of its 512 codes, yet retrieval by its word histograms reaches only 16.2 mAP (Table 2), below direct comparison of poses. Three properties of this design motivate CHOREOMATCH. The first stage discards the order of movements. Every comparison, including the edit distances, treats two words as either equal or unrelated, so neighbouring cells never match (Section A.3). And nothing records which words come from hidden body parts or lets the engine answer "unknown". The CRE keeps the order of cores, compares codes by geometry, lets only observed cores vote and rejects unindexed choreographies (Section 4.3).

D ENCODER SPECIFICATION

D.1 PER-JOINT EMBEDDING AND LOCAL BACKBONE

Each joint is embedded as

$$h^0_{t,j} = M_{t,j}\big(g([\mathbf{x}_{t-1:t+1,j}; M_{t-1:t+1,j}]) + e_j\big). \tag{11}$$

Masked attention over a neighbourhood $\mathcal{N}$ is

$$\mathrm{A}^M_{\mathcal{N}}(h)_{t,j} = \sum_{(s,j')\in\mathcal{N}(t,j)} \frac{M_{s,j'}\exp(\langle \mathbf{q}_{t,j}, \mathbf{k}_{s,j'}\rangle/\sqrt{d_h})}{\sum_{(s',j'')\in\mathcal{N}(t,j)} M_{s',j''}\exp(\langle \mathbf{q}_{t,j}, \mathbf{k}_{s',j''}\rangle/\sqrt{d_h})}\,\mathbf{v}_{s,j'}, \tag{12}$$

with head width d_h, $\mathcal{N}_{\mathrm{time}}(t,j) = \{(s,j) : |s-t| \le \delta\}$ and $\mathcal{N}_{\mathrm{body}}(t,j) = \{(t,j') : 1 \le j' \le J\}$, and the output is zero when no key in $\mathcal{N}(t,j)$ is observed. Each block applies $h \leftarrow h + \mathrm{A}^M_{\mathrm{time}}(\mathrm{LN}\, h)$, $h \leftarrow h + \mathrm{A}^M_{\mathrm{body}}(\mathrm{LN}\, h)$, $h \leftarrow h + \mathrm{FFN}(\mathrm{LN}\, h)$ and $h \leftarrow M \odot h$.

D.2 TRAINING STAGES AND OBJECTIVE

With u_b the normalized mean descriptor of performance b over its observed cores, $P(b)$ the other performances of its choreography in the batch $\mathcal{B}$, and $z'_{b,i}$ the descriptor of a rotated and rescaled copy,

$$\mathcal{L}_{\mathrm{SupCon}} = -\frac{1}{|\mathcal{B}|}\sum_b \frac{1}{|P(b)|}\sum_{p\in P(b)} \log \frac{e^{u_b^\top u_p/\tau}}{\sum_{a\neq b} e^{u_b^\top u_a/\tau}}, \tag{13}$$

$$\mathcal{L}_{\mathrm{cons}} = \operatorname*{mean}_{b,i}\big[1 - \cos(z_{b,i}, z'_{b,i})\big], \qquad \mathcal{L}_{\mathrm{uni}} = \log \operatorname*{mean}_{b\neq a} e^{-2\|u_b - u_a\|^2}. \tag{14}$$

E Additional results

Unless marked unpaired, each comparison below uses matched runs trained with identical settings on the AIST++ development split, so absolute values differ from the final model of Table 2; Table 3 lists the numbers.

E.1 Code equality against code geometry

For the same frozen encoder and codebooks, comparing codes by equality instead of by the cosine of their reconstructions lowers retrieval with a hidden body part from 88.1 to 38.0 mAP. Residual codebooks improve on a matched single codebook by 4.4 points (95% CI $[2.3, 7.0]$).

E.2 Auxiliary pose decoder

Matched runs add a pose-reconstruction decoder that differs in where it attaches, how its gradient is balanced and what it reconstructs. None improves retrieval across observation conditions; with a zero-initialized decoder, clean retrieval moves from 89.9 to 89.4, within its interval.

E.3 Joint and refined codebooks

Training the encoder and codebooks jointly with straight-through gradients changes mAP by -0.7 points (95% CI $[-2.7, 0.5]$) against a matched encoder-only control. Refining the fitted centroids with the encoder frozen, using a reconstruction loss and optionally a ranking loss, changes retrieval with a hidden body part from 88.1 to 87.9. Neither gives a gain.

E.4 Alternative anchor designs

Alternatives to the anchor of Section 4.2 encode body parts and root motion as separate input streams, or read a joint backbone into a single global output. Separate streams reduce retrieval, whereas a shared joint backbone with part and root corrections added to the ordered body readout retrieves best. The alternatives differ in more than one factor, so they motivate the design rather than isolate each correction.

E.5 Query branch

Against an anchor trained without the branch, the mean of three branch seeds raises retrieval with a hidden body part from 74.3 to 82.2 and leaves clean retrieval at 89.9 against 90.0. The comparison is unpaired and changes training as well as architecture, so it is descriptive.

E.6 Temporal alignment of codes

The CRE searches a fixed window of lags and accepts a lag only with enough observed evidence in both sequences. Replacing the single lag with subsequence dynamic-programming alignment on the same codes did not improve development retrieval.