RugBot: Lightweight Geometric Representations for Turkish Knot Sub-Action Classification

Tejvir S. Mann
University of Texas at Austin

April 2026

Abstract

Hand-knotted pile carpets tied with the symmetric Turkish (Ghiordes) knot are durable cultural artifacts, yet their production still rests on enormous amounts of skilled, *repetitive* manual work—thousands of near-identical motions per square meter—with essentially no published machine-learning or robotics pipeline devoted specifically to this craft. We decompose one complete knot cycle into seven sub-actions and ask whether a compact kinematic signal can outperform large pre-trained vision models when labeled demonstrations are scarce. We collect fifty overhead videos with per-frame human labels and approximately three hundred scripted MuJoCo episodes of an SO-101 hook on a miniature warp, encoding each frame with thirteen aligned features (hook pose, velocity, engagement, and arm/gripper channels that are fully populated in simulation but partially zeroed on real footage). Using eleven-frame sliding windows and a mixed synthetic–real training pool (≈295k windows), a small Transformer encoder reaches **test macro-F1** = 0.88 on held-out real video, exceeding SigLIP with a linear probe (validation macro-F1 = 0.67), SigLIP–kinematics fusion (validation = 0.71), and SmolVLM zero-shot (near chance on a sixty-four-frame smoke test), while training *only* on synthetic windows collapses to **real-test macro-F1** = 0.112 on the same real test split. Mixing synthetic data with real labels nevertheless raises a real-only Transformer from 0.64 to 0.88, showing that simulation helps when real anchors exist. Feature ablations expose a sim-to-real pitfall: models trained mostly on simulation lean on joint-angle features that are absent in real recordings, whereas real-only models rely on hook position and velocity. Overall, geometry-first sequence modeling offers an interpretable, data-efficient route for this specialized task compared with internet-scale visual encoders alone.

1 Introduction and Research Background

1.1 Introduction

Hand-knotted pile rugs—including carpets tied with symmetric "Turkish" (Ghiordes) knots across much of West and Central Asia—are cultural and economic artifacts that can last decades or centuries when cared for (Ford, 1981). Their surface is built from enormous repetition: each row of pile is tied knot by knot onto stretched warp threads, beaten into place, and advanced along the loom. The skill is real (spatial judgment, tension, rhythm), yet a large share of workshop time is *repetitive* manual labor: the weaver returns to the same small motion vocabulary thousands of times per square meter. That mixture—high craft embodied in mostly short, stereotyped motions—makes knot tying a compelling target for structured robot learning: if sub-actions can be sensed and classified reliably, a finite-state controller can sequence a small library of motion primitives instead of learning an uninterpretable end-to-end policy.

Turkish knot tying, also known as the Ghiordes or Azari knot, is a textile technique with origins spanning over two millennia (Eiland and Eiland, 2003; Thompson, 2006) (Figure 1). Each knot is individually tied around a pair of vertical warp threads: the yarn is looped symmetrically around both threads, and the two free ends are pulled downward between them. A skilled weaver ties thousands of these knots per day to produce a single carpet, yet the process remains entirely manual—no robotic system exists

Figure 1: A hand-knotted Turkish carpet. Each tuft in the pile is individually tied using the symmetric Ghiordes knot, thousands of times per square meter.

for automating Turkish knot tying (HALI Magazine, 2019; Brüggemann and Boehmer, 2010).

Despite progress in dexterous manipulation and rope-like objects in robotics, there is essentially no published ML pipeline devoted to *Turkish knot* tying as a labeled sub-action sequence tied to control. Large-scale approaches such as DreamDojo (Gao et al., 2026) train generalist world models on tens of thousands of hours of egocentric video, achieving broad generalization; however, these systems require billions of parameters and massive compute—resources unavailable for specialized craft tasks where demonstration data is scarce. Pretrained Vision-Language Models (VLMs) such as SigLIP (Zhai et al., 2023) and SmolVLM (Allal et al., 2025) offer an alternative via zero-shot transfer, but their training data contains no Turkish knot tying examples.

This paper asks: *Can lightweight geometric keypoint representations—trained on a small number of human demonstrations—outperform pretrained Vision-Language Models for sub-action classification in Turkish knot tying?*

We define a seven-class sub-action taxonomy (GRASP, LOOP_R, LOOP_L, PULL, BEAT, ADVANCE, RESET) decomposing one complete knot cycle. Using 50 self-recorded video clips and approximately 300 synthetic MuJoCo episodes of the SO-101 arm, we extract a compact 13-dimensional kinematic feature vector per frame and systematically compare logistic regression, LSTM, temporal convolutional, and Transformer models on kinematic features; frozen SigLIP encoders with linear and MLP heads; SmolVLM zero-shot; and multimodal fusion.

Our contributions are:

1. A novel seven-class sub-action taxonomy and labeled dataset for Turkish knot tying (50 real clips + ∼300 synthetic episodes).
2. A systematic comparison of kinematic, vision, and fusion models, finding that 13 kinematic features on a Transformer (F1 = 0.88) decisively outperform all vision approaches.
3. Feature ablation revealing that the mixed-data and real-only models exploit fundamentally different feature strategies—a key insight for sim-to-real transfer.
4. A controlled sim-to-real experiment showing synthetic augmentation boosts F1 by ≈0.24 over real-only training, while synthetic-only training collapses on the real test split (macro-F1 = 0.112).

Sub-action classification is not merely a post-hoc evaluation step. In the intended deployment pipeline (Figure 2 in §2), the predicted label selects the FSM state, which triggers the corresponding motion primitive on the robot arm. Classifier accuracy therefore directly determines task success.

1.2 Research Background

1.2.1 Imitation learning and world models

Imitation learning enables robots to acquire manipulation skills from human demonstrations. Action Chunking with Transformers (ACT) learns multi-step motor commands from a small number of teleoperated demonstrations (Zhao et al., 2023). Diffusion Policy frames action prediction as iterative denoising, generating smooth trajectories from visual observations (Chi et al., 2023). R3M learns transferable visual representations from egocentric video (Nair et al., 2022). At the frontier, DreamDojo trains a 2–14 billion parameter world model on 44,000 hours of egocentric video (Gao et al., 2026). These systems require orders of magnitude more data than is available for specialized craft tasks—motivating the lightweight approach explored here.

1.2.2 Vision-language models in robotics

Vision-language models trained on internet-scale data have been adapted for robotic perception. CLIP learns aligned visual–textual representations from 400 million image–text pairs (Radford et al., 2021). SigLIP replaces CLIP's contrastive softmax with a sigmoid loss, improving transfer (Zhai et al., 2023). The Vision Transformer (ViT) architecture (Dosovitskiy et al., 2021) underpins many modern visual encoders, including SigLIP. SmolVLM provides a compact (256M–500M parameter) alternative for video understanding (Allal et al., 2025). Vision-language-action models such as SmolVLA extend VLMs to output motor commands (Cadène et al., 2024). We test whether these models can recognize fine-grained craft sub-actions without domain adaptation.

1.2.3 Rope and knot manipulation

Nair et al. (2017) introduced the Berkeley Rope Manipulation dataset for rope straightening via self-supervised learning. Sundaresan et al. (2020) proposed dense visual descriptors for deformable object manipulation. TieBot learns diverse knots from demonstrations using point-cloud representations (Nie et al., 2024). JIGSAWS provides kinematic recordings of surgical knot-tying for skill assessment (Ahmidi et al., 2017). No dataset exists specifically for Turkish knot tying; the sub-action taxonomy, tool-based technique, and cultural context are unique to this work.

1.2.4 Geometric tracking and related work

Many manipulation pipelines infer hand pose from RGB via MediaPipe Hands (Zhang et al., 2020) or FreiHAND (Zimmermann et al., 2019). We instead track the crochet hook directly with OpenCV: ArUco markers on a custom three-fin rig, with HSV-based color fallback and gap-fill interpolation (Garrido-Jurado et al., 2014). Labels are produced with a custom frame-boundary tool (`label_boundaries.py`). Explicit 2-D hook features provide direct inductive bias for distinguishing sub-actions without estimating dense hand skeletons.

1.2.5 Sim-to-real transfer

Simulation offers scalable labeled data, but the gap between simulated and real observations remains challenging. Domain randomization improves transfer by varying visual and physical parameters (Tobin et al., 2017; Peng et al., 2018). MuJoCo provides a fast physics engine for manipulation research (Todorov et al., 2012), and MuJoCo Menagerie supplies validated robot models (Zakka et al., 2022). We quantify the contribution of synthetic data through a controlled experiment comparing real-only, synthetic-only, and mixed training regimes.

2 Research and Methods

2.1 Problem formulation

Let $x_t \in \mathbb{R}^F$ denote the per-frame feature vector (here F=13). We form a sliding window of odd width W=2k+1 centered at t, written $\mathbf{x}_{t-k:t+k} \in \mathbb{R}^{W\times F}$ (or flattened to $\mathbb{R}^{WF}$ for logistic regression). Labels are $y_t \in \{0,\dots,C-1\}$ with C=7 sub-actions (Grasp, Loop_R, Loop_L, Pull, Beat, Advance, Reset).

Conditional model. We treat sub-action prediction as multi-class classification with parameters θ:

$$P(y_t \mid \mathbf{x}_{t-k:t+k}; \theta). \tag{1}$$

Training objectives. With one-hot targets $q^{(i)}$ and model probabilities $p_\theta^{(i)} = \mathrm{softmax}(z^{(i)})$, training minimizes weighted cross-entropy:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{C-1} w_c\, q_c^{(i)}\, \log p_\theta\Big(c \mid \mathbf{x}^{(i)}\Big), \tag{2}$$

where class weights w_c are inversely proportional to training frequency to address class imbalance. The primary reported metric is **macro-F1**, which gives equal weight to each class regardless of frequency and is appropriate for imbalanced multi-class problems (Sokolova and Lapalme, 2009). Accuracy is omitted from the main comparison table because it over-weights majority classes; per-class precision and recall are available from the confusion matrix (Figure 6).

Where generalist world models learn continuous latent action spaces from millions of video frames (Gao et al., 2026), we learn a structured seven-class vocabulary from a comparatively small dataset—trading generality for data efficiency and interpretability.

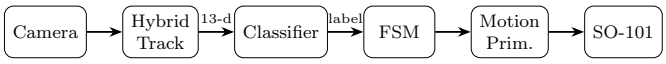

Figure 2: RugBot pipeline. The classifier prediction selects the FSM state and triggers the corresponding motion primitive. Higher accuracy reduces incorrect primitive triggers and improves simulated tying behavior.

2.2 End-to-end pipeline (perception to motion)

Figure 2 summarizes the operational control stack: sensing and tracking produce the kinematic (and optionally visual) inputs; the classifier chooses a subaction; an FSM commits to motion primitives; the simulated SO-101 arm executes the knot cycle. The predicted label is the **control interface**, not a post-hoc annotation for analysis alone.

2.3 FSM control architecture

The classifier's predicted label drives the simulated robot: each FSM state maps one-to-one to a scripted motion primitive (joint-angle waypoint sequences). Transitions require sustained confidence over multiple frames, reducing single-frame jitter. The cyclic state structure is shown in Figure 3 (§3).

2.4 Kinematic models (Group A)

We evaluate four architectures in increasing order of temporal expressivity, so that results isolate the contribution each modeling choice provides.

Logistic regression. As an interpretable lower bound, we test whether the kinematic feature space is linearly separable. Per-frame (13-d) and windowed (143-d flattened) variants use scikit-learn with `class_weight='balanced'` and `StandardScaler`; the gap between them isolates the value of temporal context without any learned dynamics.

LSTM. The LSTM tests whether recurrent sequential processing outperforms windowed-parallel approaches. LSTMs compress a history of input frames into a hidden state, making them the standard architecture for sequential time-series. Two-layer LSTM with 192 hidden units, input LayerNorm (Ba et al., 2016), mean-pooled hidden states for classification.

Temporal Conv1D. The Conv1D tests whether local temporal patterns within a few adjacent frames are sufficient, without the sequential bottleneck of recurrence. Two 1D convolutional layers (64 and 128 channels, kernel size 3) with BatchNorm, ReLU, and global average pooling.

Transformer encoder. Following Vaswani et al. (2017), we apply a linear embedding to d_{model}=128, then two encoder layers with four attention heads and feedforward dimension 256. Classification uses the center token (position k for W=11), which attends to both past and future context. The Transformer is the primary kinematic model: its bidirectional self-attention makes it the most expressive of our four architectures.

All neural kinematic models use Adam (lr = 0.001), batch size 512, 12 epochs, MPS backend.

2.5 Vision baselines (Group C)

These baselines test whether pretrained visual representations from internet-scale data encode the subaction-discriminative signal, even without domain-specific training. We use *frozen* encoders because 50 real clips yield roughly 11K labeled frames—far too few to stably fine-tune a 200M-parameter visual model without overfitting.

SigLIP + linear / MLP. Frozen SigLIP (`google/siglip-base-patch16-224`) (Zhai et al., 2023) yields 768-dimensional vectors per real RGB frame; scikit-learn trains a linear probe and a two-layer MLP head. Because RGB is only available for real clips, these rows use ∼3.5k labeled frames and report **validation** macro-F1.

SmolVLM zero-shot. SmolVLM2-256M-Video-Instruct (Allal et al., 2025) is prompted for the subaction label without any task-specific training. This is a 64-frame smoke test—not a full evaluation—designed to probe the zero-shot upper bound for internet-pretrained VLMs on this domain.

Fusion. Concatenation of 768-d SigLIP + 13-d kinematics (781-d) with a logistic regression head (∼3.75k training frames). This tests whether vision and kinematics are complementary—each encoding information the other misses—in which case fusion should outperform either modality alone.

2.6 Sim-to-real experiment (Group E)

With only 50 real clips available, simulation offers a scalable path to expand the training distribution. The question is whether synthetic features transfer usefully or whether the feature mismatch between domains causes the model to learn signals absent at test time. The Transformer trains under three regimes: (1) synthetic-only (∼284k windows, tested on real); (2) real-only (∼11k windows); (3) mixed (∼295k windows). Architecture and hyperparameters are fixed; only the training mixture changes.

2.7 Feature ablation

With 13 features spanning position, velocity, orientation, engagement, and arm joints, it is not obvious which signals drive performance. Ablation answers this: we zero one feature group at a time, retrain the Transformer from scratch with identical hyperparameters, and measure the F1 drop. We run ablations on **both** mixed and real-only splits, because the importance ranking reverses between them.

2.8 Multi-seed variance

The Transformer is trained with seeds $\{0, 1, 2\}$ to report mean ± spread on test macro-F1.

2.9 Evaluation metrics

We report macro-F1 on held-out **real** test windows for kinematic models trained on mixed data. Vision rows marked with * use validation macro-F1 on real frames only. Confusion structure is discussed via the row-normalized matrix (Figure 6), from which per-class recall and precision can be derived.

3 Materials and Data Sources

This section defines the sub-action vocabulary, how real and synthetic frames are collected, and the shared 13-dimensional feature contract used throughout §2–4.

3.1 Sub-action taxonomy

We decompose one Turkish knot cycle into seven sub-actions forming a cyclic FSM (Figure 3): **GRASP** (pick up hook), **LOOP_R** (loop yarn right around right warp thread), **LOOP_L** (loop yarn left), **PULL** (pull both yarn tails downward),

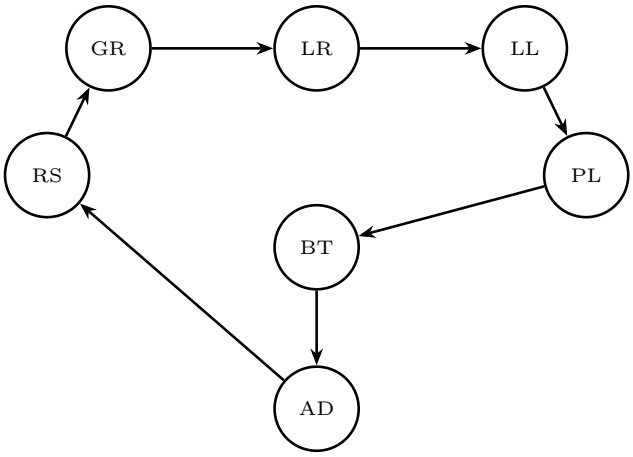

Figure 3: FSM for the Turkish knot cycle. GR=GRASP, LR=LOOP_R, LL=LOOP_L, PL=PULL, BT=BEAT, AD=ADVANCE, RS=RESET.

BEAT (comb presses knot against previous row), **ADVANCE** (reposition to next knot), and **RESET** (return hook to rest). The taxonomy follows traditional technique descriptions (Eiland and Eiland, 2003); each sub-action maps to one simulation motion primitive.

3.2 Real demonstrations

We recorded 50 video clips of a single demonstrator tying Turkish knots from a fixed overhead camera at 20 fps (compressed from 30 fps via FFmpeg). The hook was instrumented with a hybrid tracking rig: three ArUco-marked fins at 120° intervals on the shaft, plus orange electrical tape as a color fallback (Figure 4).

Hook position was extracted per frame via: (1) ArUco detection; (2) orange-tape HSV fallback; (3) linear interpolation for gaps ≤5 frames. A representative clip (335 frames) achieved 87.2% coverage: 43.6% ArUco, 40.9% orange tape, 2.7% interpolation, 15.5% untracked. Clips below 80% coverage were excluded. Hook coordinates were normalized to the warp-frame bounding box to $[0, 1]$. Clips split by filename: 35 train, 5 validation, 10 test. Labels are human-authored via a custom OpenCV tool (`label_boundaries.py`).

3.3 Synthetic data

We generated 299 synthetic episodes using a MuJoCo simulation of the SO-101 robot arm (Todorov et al., 2012; Zakka et al., 2022) (Figure 5). The

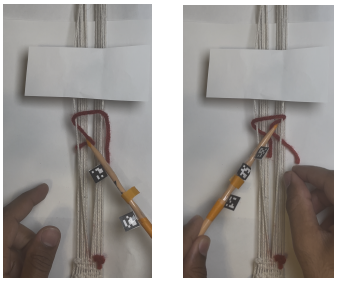

Figure 4: Real recording setup. ArUco marker fins and orange tape are visible on the hook; red yarn loops around white warp threads.

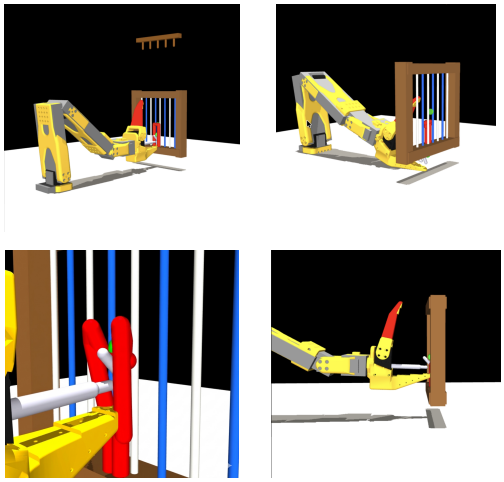

Figure 5: MuJoCo simulation. Top: wide views of SO-101 arm at the warp frame. Bottom: close-ups of the hook engaging with warp threads and yarn.

scene includes the SO-101 with a custom hook end-effector, a warp frame, and a scripted yarn spline updated by the FSM.

Labels are assigned deterministically by FSM state (zero manual labeling). Per-episode joint noise ($\sigma = 0.04$ rad) discourages memorization of fixed time series; feature noise ($\sigma = 0.003$) is added at recording time. Episodes are validated automatically: `hook_engaged` must activate during loops, gripper closure must occur during PULL, and outliers are filtered. The combined training pool yields 295,186 training windows (283,968 synthetic + 11,218 real), with one window generated per labeled frame using W=11 sliding windows.

3.4 Feature construction

Both domains produce an identical 13-dimensional feature vector per frame (Table 1), enabling direct mixing. Hook orientation (d_x, d_y) and all five arm joint angles are sim-only (zeroed in real clips), so seven of thirteen dimensions are unavailable at real test time—a key source of the sim-to-real gap discussed in §4.

For sequence models we use sliding windows of width $W = 11$ centered on each target frame with edge padding ([11 × 13] tensors; 143-d vectors for windowed logistic regression).

3.5 Auxiliary context

Related rope benchmarks (e.g. Berkeley rope manipulation (Nair et al., 2017)) contextualize deformable-object learning but are not used as training inputs; all reported models train on the self-collected split unless explicitly stated.

4 Results

4.1 Model Comparison

Table 2 presents the main results. The Transformer achieves the highest test macro-F1 of 0.880, with Conv1D nearly identical (0.878). All kinematic models except the LSTM outperform all vision approaches.

Logistic regression per-frame (0.791) shows that even instantaneous kinematic snapshots carry substantial class information: the 13-dimensional feature vector, encoding arm configuration and hook state at a single moment, is already partially linearly separable across the seven classes. Adding the 11-frame window raises logistic regression by 6.4 points (0.855), confirming that temporal context—the direction and rate of motion across frames—is needed to resolve ambiguous boundary frames where two sub-actions share similar instantaneous poses.

Conv1D and the Transformer reach near-identical performance (0.878 vs. 0.880). This parity reveals that the discriminative temporal signal in an 11-frame kinematic window is largely *local*: captured well by Conv1D's kernel-3 filters, and not requiring the long-range pairwise attention of a Transformer. What matters for classification is the motion pattern across a few neighboring frames, not relationships between frames far apart in the window.

The LSTM collapses to F1 = 0.134 despite using the same data as Conv1D and Transformer. LSTMs build their representation by propagating a hidden state sequentially through the input; on a short fixed window of only 11 frames, the recurrent state has

Table 1: The 13 kinematic features per frame. Features marked † are zero in real clips (sim-only channels).

Feature	Dim	Real?
Hook position (x, y)	2	✓
Hook velocity (v_x, v_y)	2	✓
Hook orientation $(d_x, d_y)^\dagger$	2	—
Hook engaged	1	✓
Gripper open†	1	—
Arm joints $(j_0$–$j_4)^\dagger$	5	—

Table 2: Model comparison. Kinematic models train on mixed data; vision models use real clips only; * = validation F1 (test-set frames unavailable for vision rows). "Train windows" are post-windowing sample counts (W=11). Best in **bold**.

Model	Input	Train windows	F1
Kinematic (Group A)			
LogReg (per-frame)	13-d	295K	0.791
LogReg (window-11)	143-d	295K	0.855
LSTM	[11,13]	295K	0.134
Conv1D	[11,13]	295K	0.878
Transformer	[11,13]	295K	**0.880**
Vision (Group C)			
SigLIP + Linear	768-d	3.5K	0.670*
SigLIP + MLP	768-d	3.5K	0.643*
Fusion (SigLIP+kin)	781-d	3.8K	0.709*
SmolVLM zero-shot	RGB	—	0.035†
Sim-to-real (Group E)			
Transformer (sim-only)	[11,13]	284K	0.112

†64-frame smoke test only; not a full test-set evaluation.

too few steps to converge to a useful representation, and mean-pooling the resulting hidden states averages out whatever temporal structure was captured. Parallel architectures (Conv1D, Transformer) process the entire window simultaneously and are not subject to this sequential instability.

Among vision models, SigLIP + Linear (0.670 val) is the best single-modality vision result. This is substantially below the kinematic Transformer (0.880) despite SigLIP being a much larger model trained on far more data—the gap reflects inductive bias mismatch. SigLIP encodes general visual semantics; the discriminative signal here is geometric (hook direction, arm configuration, motion trajectory), not appearance. LOOP_R and LOOP_L, for example, look nearly identical in a single frame yet are easily separated by kinematic velocity.

Fusion (0.709) improves over vision-only by incorporating kinematic features, but remains 17 points below kinematics-only. The gap exists for two reasons: first, the fusion model is constrained to ∼3.75k real frames because SigLIP features cannot be generated for synthetic episodes without rendering MuJoCo visual output; second, adding 768 noisy visual dimensions to 13 discriminative kinematic dimensions dilutes rather than complements the geometric signal.

SmolVLM (0.035 on a 64-frame smoke test) fails entirely. SmolVLM's training corpus contains no Turkish knot tying, and the visual appearance of hook, yarn, and warp does not overlap enough with its training distribution for zero-shot reasoning to succeed.

Per-class F1 is highest for LOOP_R (0.96), BEAT (0.95), and LOOP_L (0.95) and lowest for GRASP (0.77) and RESET (0.74). Figure 6 shows the row-normalized confusion matrix, where diagonal entries are per-class recall: LOOP_R (96.6%), BEAT (96.3%), LOOP_L (93.5%), while GRASP (79.6%) and PULL (79.2%) show the most recall error. The dominant off-diagonal entries are PULL→GRASP (15.1%) and RESET→GRASP (15.0%)—both in-

volve the hand returning toward the hook's resting position, sharing similar arm configurations with GRASP.

4.2 Feature Ablation: Mixed vs. Real-Only

Table 3 compares ablations on mixed data (295K windows) and real-only data (11K windows), revealing that the two models exploit fundamentally different feature strategies.

Mixed data. Arm joint angles are the single most important feature group ($\Delta = -0.132$), more than double the impact of removing all hook features combined ($\Delta = -0.051$). This is explained by the training distribution: 96% of the 295K training windows are synthetic, and synthetic episodes have fully populated arm joint values. The mixed-data model correctly learns to lean on joint configuration because it is a rich, reliable signal in the dominant portion of its training data. Hook velocity ($\Delta = -0.035$) matters more than hook position ($\Delta = -0.005$), consistent with the intuition that motion dynamics are more diagnostic of sub-action identity than static location.

Real-only data. The hierarchy inverts entirely. Arm joints—always zero in real clips—have negligible impact ($\Delta = -0.017$). Hook position ($\Delta = -0.069$) and hook velocity ($\Delta = -0.073$) become the dominant signals, as these are the only richly populated features in real recordings. Removing all hook kinematics causes total collapse to F1 = 0.045 (the model predicts nearly only BEAT). Hook engagement removal actually improves performance by 1.3 points, suggesting this binary heuristic adds noise in the real-only regime.

Interpretation. The 23.6-point gap between mixed (0.880) and real-only (0.644) is not purely data volume. It reflects the model's access to seven sim-only feature dimensions that are present and informative in mixed training but absent at real test time—a performance that does not guarantee robust real-world generalization.

4.3 Effect of Synthetic Data

Table 4 compares training regimes. Synthetic augmentation nearly doubles real-only performance (0.644 → 0.880). Synthetic-only training fails (F1 = 0.112) because the features most relied upon in simulation (arm joints, hook orientation) are zeroed in real test data. Mixed training succeeds because real examples anchor the model to features shared across both domains—hook position and velocity—while synthetic examples provide the volume needed to fit those features reliably.

Comparing real-only performance across architectures: LogReg per-frame 0.409, LogReg windowed 0.546, Conv1D 0.614, Transformer 0.644. All degrade substantially from their mixed-data scores, but the Transformer's 3-point edge over Conv1D on 11K real windows suggests that model capacity helps when labeled data is scarce.

4.4 Multi-Seed Stability

Across three random seeds, the Transformer achieves test F1 of 0.877 ± 0.003 (range: 0.874–0.879), confirming stability across initializations. RESET is consistently the weakest class (F1: 0.731–0.743 across seeds).

5 Discussion and Conclusion

5.1 Summary of Findings

We asked whether lightweight geometric representations could outperform pretrained VLMs for Turkish knot sub-action classification. The answer is clear: a Transformer on 13 kinematic features (F1 = 0.880) outperforms the best vision baseline by 21 points. Three findings emerge:

1. **Geometric features outperform visual encoders.** For this craft task, 13 kinematic features encode discriminative signal more efficiently than 768-dimensional visual features from hundreds of millions of internet images.
2. **Synthetic augmentation is valuable but not sufficient alone.** Adding 283,968 synthetic training windows boosts F1 by ≈0.24 over real-only (mixed 0.880 vs. real-only 0.644), but synthetic-only training yields real-test F1 = 0.112, revealing a domain gap rooted in feature misalignment.
3. **Mixed and real-only models use different features.** The mixed model relies on sim-only arm joints ($\Delta = -0.132$); the real-only model relies on hook position/velocity ($\Delta \approx -0.07$ each). This divergence is a key insight for sim-to-real transfer.

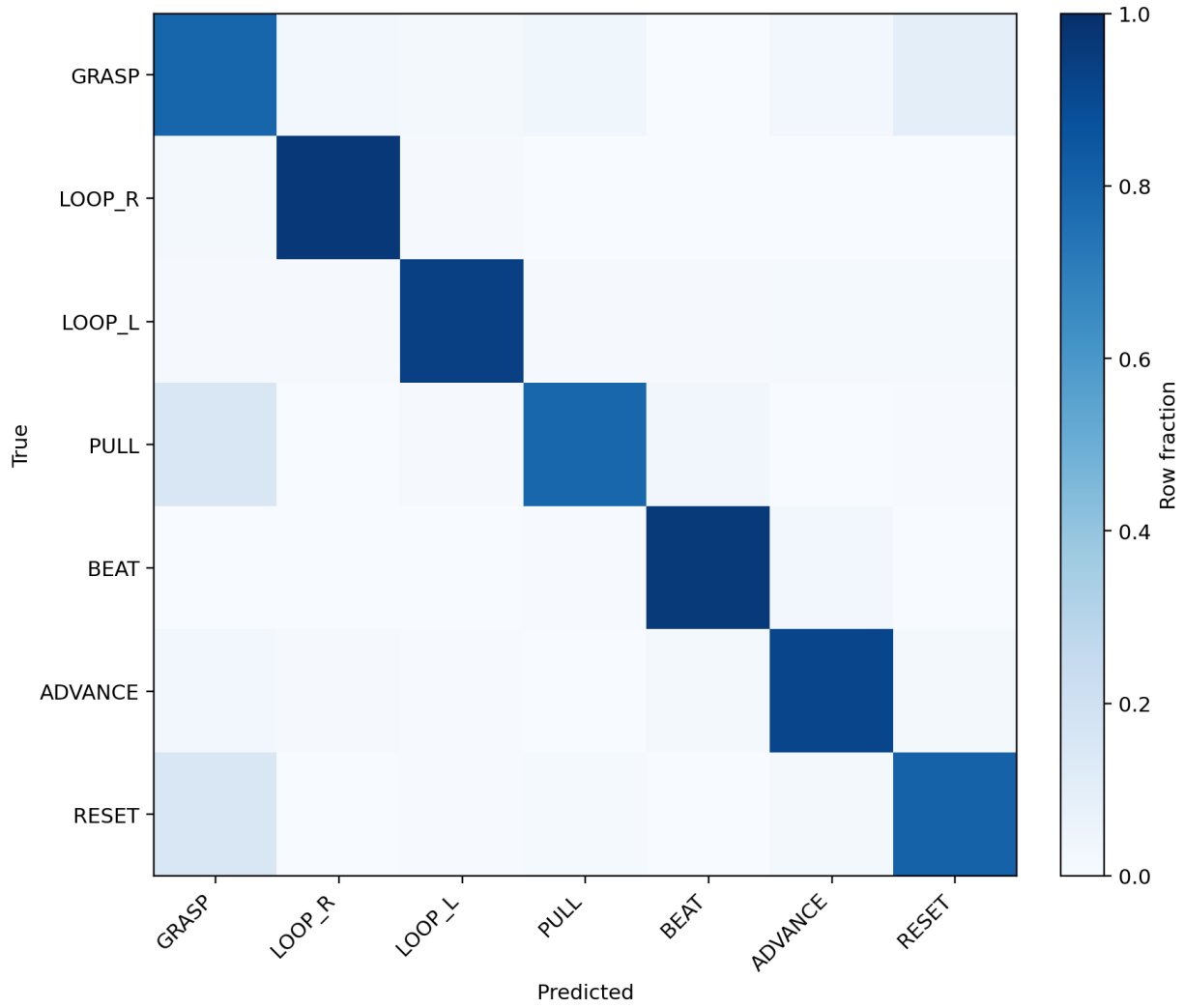

Figure 6: Row-normalized confusion matrix (Transformer, mixed test set). Diagonal entries are per-class recall. LOOP_R and BEAT exceed 96%; LOOP_L reaches 93.5%. The two largest off-diagonal entries—PULL→GRASP (15.1%) and RESET→GRASP (15.0%)—reflect similar arm configurations during transitional motions.

5.2 Why Kinematics Outperforms Vision

The seven sub-actions are defined by hook geometry relative to the warp, not visual appearance. LOOP_R and LOOP_L look visually similar—same hook, same yarn—but differ in hook direction, captured directly by velocity features. SigLIP is optimized for internet image semantics, not tool-pose discrimination.

The comparison is confounded by data volume: kinematic models train on 295K frames while vision models are limited to 3,500 real frames (SigLIP features cannot be generated for synthetic episodes without rendering MuJoCo visual output). However, the magnitude of the gap (21 points) and Conv1D's near-equivalent performance to the Transformer suggest that the kinematic feature space is fundamentally better suited to this task, not merely better served by more data.

5.3 Error Analysis

The dominant errors—PULL→GRASP (15.1%) and RESET→GRASP (15.0%)—are physically meaningful. PULL and RESET both involve the hand returning toward the hook's resting position, sharing similar arm configurations with GRASP. These confusions are concentrated at sub-action boundaries where the motion is transitional. The high reliability of LOOP_R (96.6%), BEAT (96.3%), and LOOP_L (93.5%) suggests that mid-action frames, where motion is most distinctive, are easily classified.

Table 3: Feature ablation on mixed vs. real-only data. The mixed model relies on arm joints (sim-only); the real-only model relies on hook position and velocity. Removing all hook kinematics on real-only causes total collapse.

	Mixed		Real-only	
Ablation	F1	Δ	F1	Δ
Full baseline	.880	—	.644	—
No hook pos.	.875	−.005	.575	−.069
No hook orient.	.877	−.004	.642	−.002
No hook engag.	.869	−.011	.657	+.013
No hook vel.	.845	−.035	.571	−.073
No hook kin. (all)	.829	−.051	.045	−.599
No arm joints	.749	−.132	.627	−.017

Table 4: Training regime comparison (Transformer). All rows use the same architecture and hyperparameters; only the data source changes. Test F1 is macro-F1 on the held-out real test split.

Regime	Train windows	Test F1
Real only	11,218	0.644
Synthetic only	283,968	0.112
Mixed	295,186	0.880

5.4 Limitations

Several limitations qualify these findings. (1) All 50 clips come from a single demonstrator; generalization to other weavers is untested. (2) A single annotator labeled all clips without inter-rater reliability. (3) The kinematic–vision comparison is confounded by data volume (295K vs. 3,500 frames). (4) The simulation uses a scripted yarn spline, not true rope physics. (5) No closed-loop FSM evaluation was conducted on physical or simulated hardware. (6) The LSTM's collapse to F1 = 0.134 is consistent with its architectural mismatch to short fixed windows, but further hyperparameter tuning may recover some performance.

5.5 Future Work

Three directions follow. First, deploying the classifier on a physical SO-101 arm would validate the pipeline end-to-end; the domain gap motivates fine-tuning with real robot episodes. Second, Turkish knot tying is fundamentally bimanual; AnyBimanual (Lu et al., 2025) provides a plug-and-play transfer path from our unimanual pipeline to a two-arm system. Third, rendering visual frames from MuJoCo would enable a data-controlled comparison between kinematic and vision features, testing whether vision's underperformance is intrinsic or reflects data starvation. Additionally, training models exclusively on domain-shared features (hook position, velocity, engagement) could improve sim-to-real transfer robustness by preventing reliance on sim-only signals.

5.6 Conclusion

Hand-knotted carpets endure because the underlying motions are disciplined and repeatable; the same structure makes knot tying amenable to explicit sub-action models. This work shows that, on our dataset, compact kinematics plus a small sequence model outperform internet-scale vision encoders for phase classification—while also warning that mixed sim–real training can hide reliance on simulation-only channels. Closing that gap (shared features, rendered RGB, hardware loops, and bimanual policies) is the natural next step toward robots that assist rather than replace the craft.

References

Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Hager, G., et al. 2017. A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Transactions on Biomedical Engineering*, 64(9):2025–2041.

Allal, L. B., Sanh, V., Lhoest, Q., et al. 2025. SmolVLM: Small vision language models. *HuggingFace Blog*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Brüggemann, W. and Boehmer, H. 2010. *Encyclopaedia of Oriental Rugs*. Hali Publications.

Cadène, R., Alibert, S., Soare, A., et al. 2024. LeRobot: State-of-the-art machine learning for real-world robotics in PyTorch. *GitHub repository*.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of RSS*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of ICLR*.

Eiland, M. L. and Eiland, M. L. 2003. *Oriental Carpets: A Complete Guide*. Bulfinch Press, 5th edition.

Ford, P. R. J. 1981. *Oriental Carpet Design: A Guide to Traditional Motifs, Patterns and Symbols*. Thames and Hudson.

Gao, S., Liang, W., Zheng, K., et al. 2026. DreamDojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292.

HALI Magazine. 2019. The art and craft of the hand-knotted carpet. *HALI: The International Journal of Oriental Carpets and Textiles*, 201.

Li, K., Li, P., Liu, T., Li, Y., and Huang, S. 2025. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of CVPR*.

Lu, G., Yu, T., Deng, H., Chen, S. S., Tang, Y., and Wang, Z. 2025. AnyBimanual: Transferring unimanual policy for general bimanual manipulation. In *Proceedings of ICCV*.

Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., and Levine, S. 2017. Combining self-supervised learning and imitation for vision-based rope manipulation. In *Proceedings of ICRA*.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. 2022. R3M: A universal visual representation for robot manipulation. In *Proceedings of CoRL*.

Nie, Y., Zhou, Y., Wu, J., et al. 2024. TieBot: Learning to knot a tie from visual demonstrations. In *Proceedings of ICRA*.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *Proceedings of ICRA*.

Radford, A., Kim, J. W., Hallacy, C., et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of ICML*.

Sokolova, M. and Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Sundaresan, P., Grannen, J., Thananjeyan, B., Balakrishna, A., Laskey, M., Stone, K., Gonzalez, J. E., and Goldberg, K. 2020. Learning rope manipulation policies using dense object descriptors trained on synthetic depth. In *Proceedings of ICRA*.

Thompson, J. 2006. *Carpet Magic: The Art of Carpets from the Tents, Cottages and Workshops of Asia*. Barrie & Jenkins.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *Proceedings of IROS*.

Todorov, E., Erez, T., and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *Proceedings of IROS*.

Vaswani, A., Shazeer, N., Parmar, N., et al. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Zakka, K., Tassa, Y., and the MuJoCo Team. 2022. MuJoCo Menagerie: A collection of validated robot models. *GitHub repository*.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *Proceedings of ICCV*.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. 2020. MediaPipe Hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. 2023. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of RSS*.

Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., and Brox, T. 2019. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *Proceedings of ICCV*.

Appendix: AI Usage Statement

AI tools were used for grammar and spelling corrections and for brainstorming an initial thesis statement. All experimental design decisions—including the sub-action taxonomy, data collection protocol, model architectures, and evaluation methodology—were made independently by the author. All 50 real video clips were recorded and hand-labeled by the author using a custom OpenCV tool. All result numbers reported in this paper come from experiments run by the author on local hardware.