NOV 6-9, 2024
Munich, Germany

InterACT: Inter-dependency Aware Action Chunking with Hierarchical Attention Transformers for Bimanual Manipulation

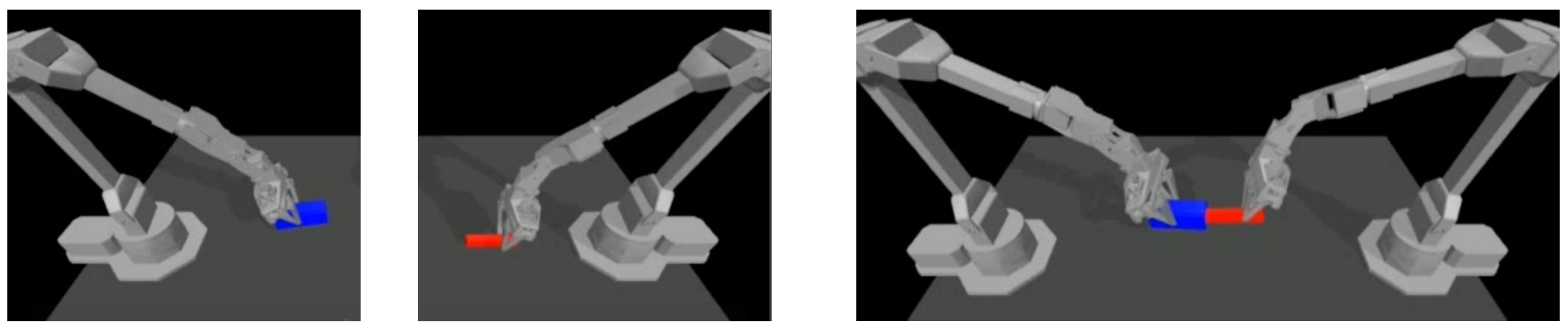
Andrew Lee[1], Ian Chuang[1,2], Ling–Yuan Chen[1], Iman Soltani[1]
[1]University of California Davis, [2]University of California Berkeley

SCAN FOR FULL PAPER AND VIDEOS

InterACT

- ❑ **InterACT** is a model **designed for bimanual manipulation** tasks
- ❑ **InterACT** leverages **hierarchical attention to capture inter-dependencies** between joint states and visual features
- ❑ **InterACT** is capable of handling **varying levels of coordination** within bimanual manipulation tasks
- ❑ **InterACT outperforms baseline** ACT in both simulation and real-world bimanual manipulation tasks

1. Motivation



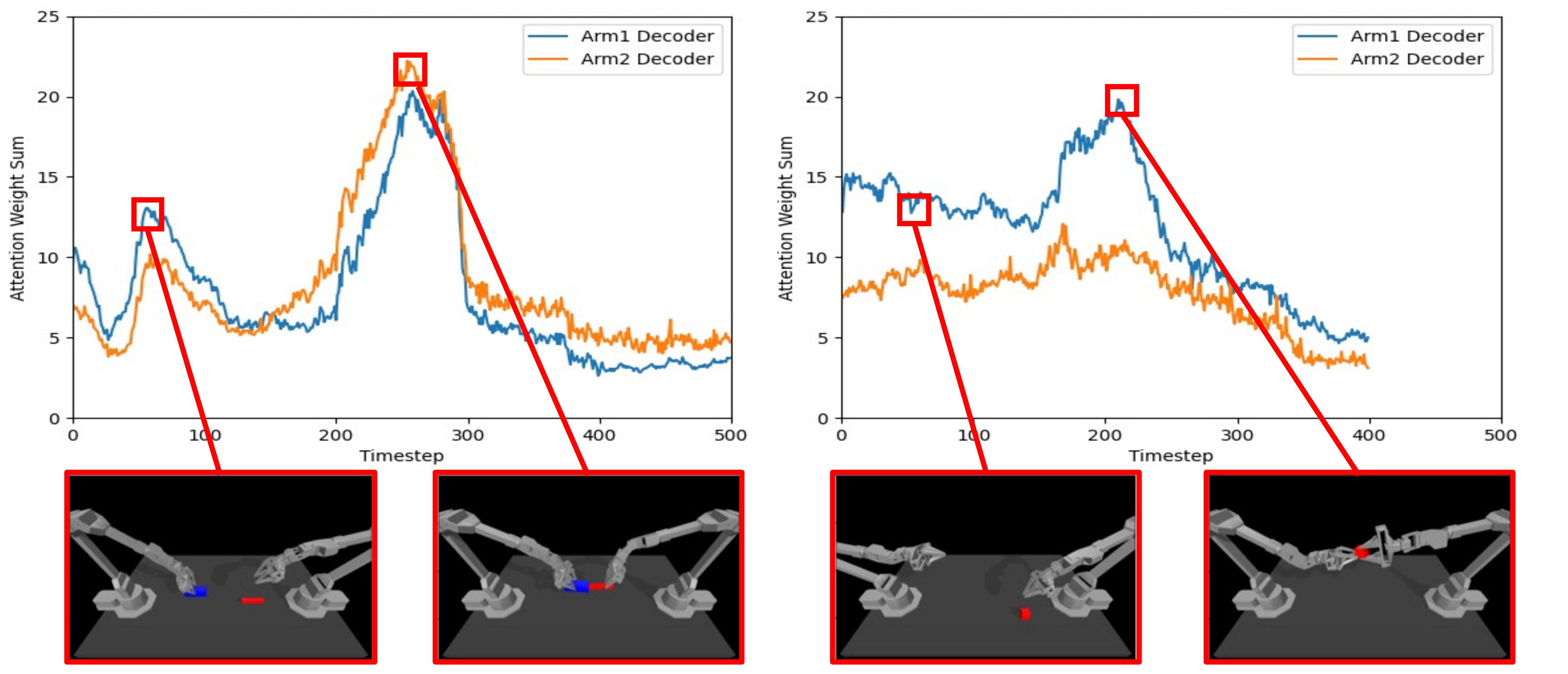
Low levels of coordination High levels of coordination

- ❑ Bimanual Manipulation tasks rely on **varying levels of coordination** between the two arms at different phases of the task
- ❑ An effective model should be able to handle virtually independent movement of each arm as well as heavily coordinated actions

2. Solution

- ❑ We propose **InterACT** designed for bimanual tasks
  - ○ InterACT **leverages hierarchical attention to capture inter-dependencies** between dual-arm joints and visual input
  - ○ InterACT decouples action prediction for two arms with parallel decoders
- ❑ Key components of **InterACT**
  - ○ **CLS tokens** — special tokens represent and summarize each input segment, facilitating better coordination between arms
  - ○ **Hierarchical Attention Encoder:** consists two components
    1) Segment-Wise Encoder — captures **intra-dependencies** within each arm's joint and visual features
    2) Cross-Segment Encoder — captures **inter-dependencies** across arms and visual features
  - ○ **Multi-arm Decoder:** consists two components
    1) Arm-specific decoders — two parallel decoders generate independent actions for each arm
    2) Synchronization block — share information across decoders to ensure coordinated actions

3. Key Results



Attention weights for **other arm's CLS tokens** at decoder (left: insert peg, right: cube transfer)

- ❑ CLS tokens play a crucial role during the coordination phase
  - ○ CLS tokens that captured inter-dependencies give the model flexibility to leverage the information when needed

	Transfer Cube (Sim)			**Peg Insertion** (Sim)			**Slide Ziploc** (Real)			**Thread Velcro** (Real)		
	Touch	Lift	**Transfer**	Grasp	Contact	**Insert**	Grasp	Pinch	**Open**	Lift	Grasp	**Insert**
ACT	82	60	50	76	66	20	96	92	88	88	42	16
InterACT	**98**	**88**	**82**	**88**	**78**	**44**	96	92	**92**	**94**	**56**	**20**

	Slot insertion (Sim)		**Insert Plug (Real)**		**Click Pen (Real)**		**Sweep (Real)**		**Unscrew cap (Real)**	
	Lift	**Insert**	Grasp	**Insert**	Grasp	**Click**	Grasp	**Sweep**	Touch	**Unscrew**
ACT	96	88	92	30	92	56	88	42	84	60
InterACT	**100**	**100**	92	**42**	**94**	**62**	**92**	**52**	**88**	**62**

Results on simulated and real-world tasks

- ❑ **InterACT** outperforms baseline ACT on bimanual tasks — in all low and high level of coordination phases

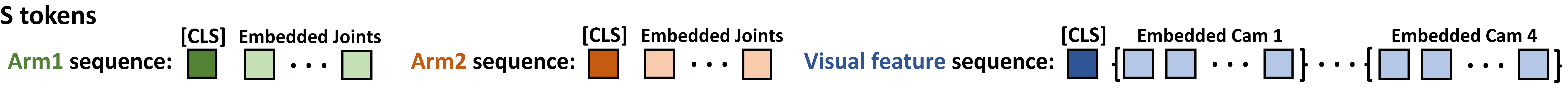
	Transfer Cube			**Peg Insertion**			**Slot Insertion**	
	Touch	Lift	**Transfer**	Grasp	Contact	**Insert**	**Lift**	Insert
InterACT (no CLS Tokens)	98	84	84	70	68	22	100	86
InterACT (no CS Encoder)	80	72	72	84	80	24	100	98
InterACT (no Sync Block)	74	54	54	**90**	**86**	30	100	100
InterACT (all components)	**98**	**88**	**84**	88	78	**44**	**100**	**100**

Ablation study results

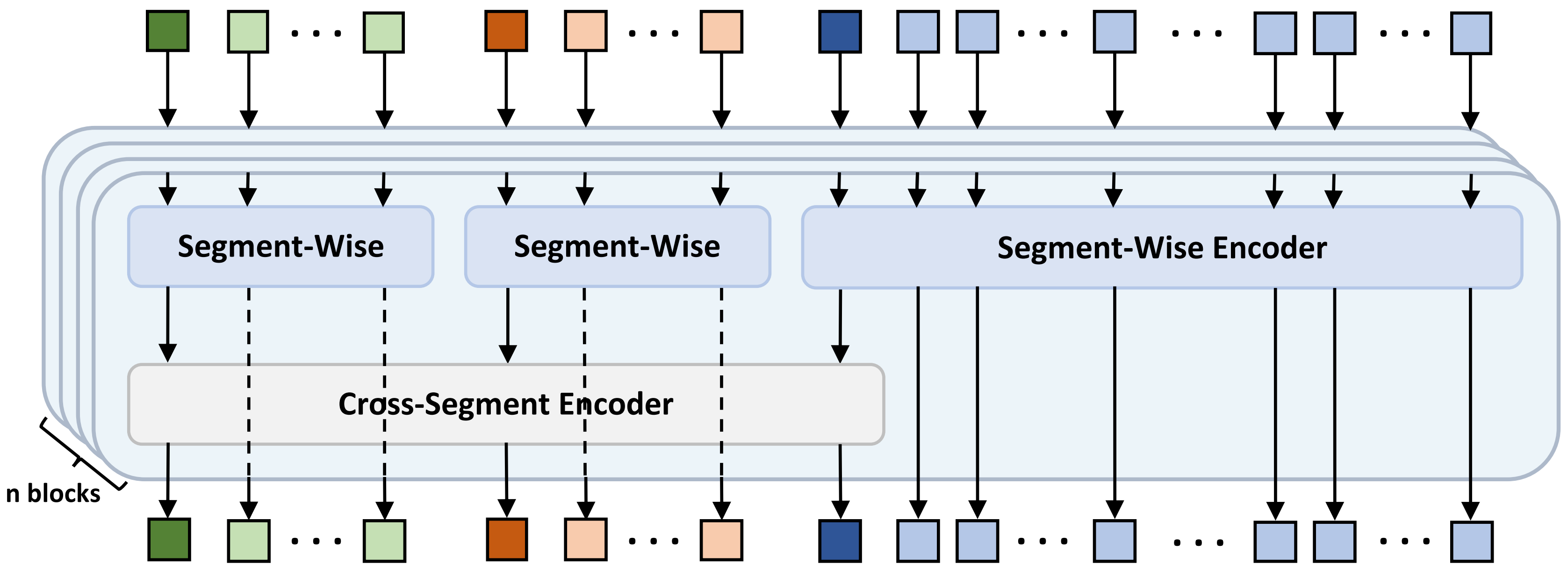
- ❑ Ablation demonstrates our components — **CLS tokens, Cross-Segment Encoder, and Synchronization block** contributes to better bimanual manipulation performance when utilized together

4. Architecture of **InterACT**

❑ **CLS tokens**

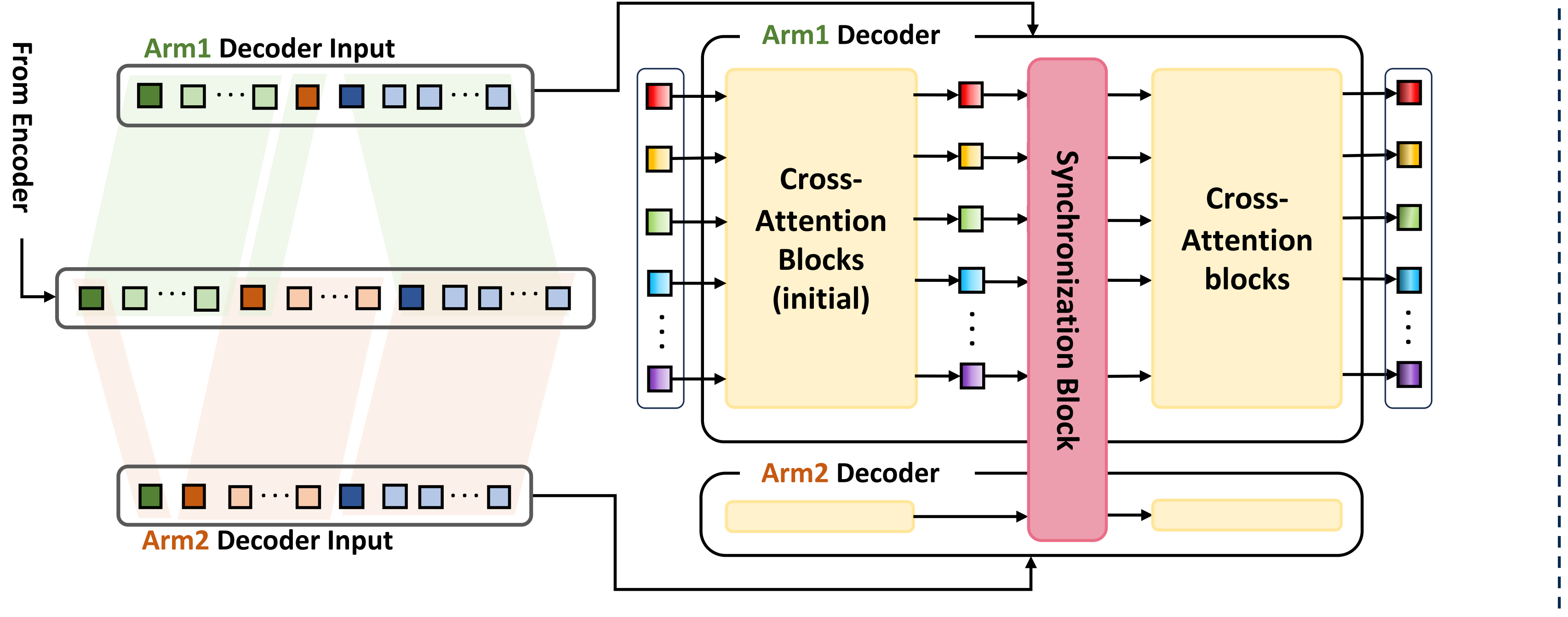


❑ **Hierarchical Attention Encoder**



- ❑ **Segment-Wise Encoder**
  - ○ Process each segment (two joints, visual data) independently
  - ○ CLS tokens capture **intra-dependency** information
- ❑ **Cross-Segment Encoder**
  - ○ Process only **CLS tokens** from the segment-wise encoders
  - ○ CLS tokens capture **inter-dependency** information

❑ **Multi-arm Decoder**



- ❑ **Arm-specific decoders:** responsible for actions for its respective arm
  - ○ Input only relevant tokens — **its arm sequence, other arm's CLS tokens and visual features**
- ❑ **Synchronization block**
  - ○ Process Intermediate output from initial cross-attention blocks of the two decoders