ConceptACT: Integrating High-Level Semantic Concepts into Transformer-Based Imitation Learning

Jakob Karalus
Institute of Artificial Intelligence
Ulm University
Ulm, Germany
jakob.karalus@uni-ulm.de

Abstract

Imitation learning in robotics allows humans to teach complex tasks by demonstration. While this training regime is quite powerful, most current approaches only rely on directly recorded data, such as joint values and image inputs. In this work we address this limitation by allowing humans to provide high-level annotations for each episode, which can contain additional semantic information. We include this information in the training process through a concept transformer and therefore enforce that the learning model can handle this additional information during training. We show in an experiment involving "pick and place" with additional sorting constraints that our extension of the ACT architecture (which we call ConceptACT) can lead to faster learning performance. Specifically, ConceptACT achieves a significant reduction in optimality gap compared to standard ACT, demonstrating that properly integrated semantic concepts can significantly improve sample efficiency in robotic imitation learning.

1 Introduction

Imitation learning has emerged as a powerful paradigm for teaching robots complex manipulation skills by directly learning from human demonstrations. Rather than designing explicit reward functions or control algorithms, this approach enables robots to acquire behaviors by observing expert trajectories, making it particularly valuable for tasks where success criteria are easier to demonstrate than to formalize. However, current imitation learning methods suffer from a fundamental limitation: they rely exclusively on low-level sensorimotor data (joint positions, images, forces) while ignoring the rich semantic knowledge that humans naturally use when teaching and learning tasks.

When humans teach complex tasks to other humans, they instinctively employ a variety of scaffolding techniques—providing conceptual frameworks, highlighting important features, and explaining underlying principles—to accelerate the learning process. In contrast, machine learning from demonstration typically operates as a black box, learning implicit associations between high-dimensional observations and actions without access to the semantic reasoning that guides human decision-making. This mismatch represents a significant missed opportunity: human demonstrators possess valuable high-level knowledge about task structure, object properties, and causal relationships that could substantially improve learning efficiency if properly leveraged.

Recent advanced in Imitation Learning allowed the learning of complex locomotor tasks quite quickly. However, even state-of-the-art approaches like ACT remain limited to learning from raw sensorimotor streams, failing to exploit the conceptual understanding that humans can readily provide alongside their demonstrations. This limitation is especially pronounced in manipulation tasks involving complex reasoning about object properties, spatial relationships, or task constraints—precisely the scenarios where semantic guidance would prove most beneficial.

18th European Workshop on Reinforcement Learning (EWRL 2025).

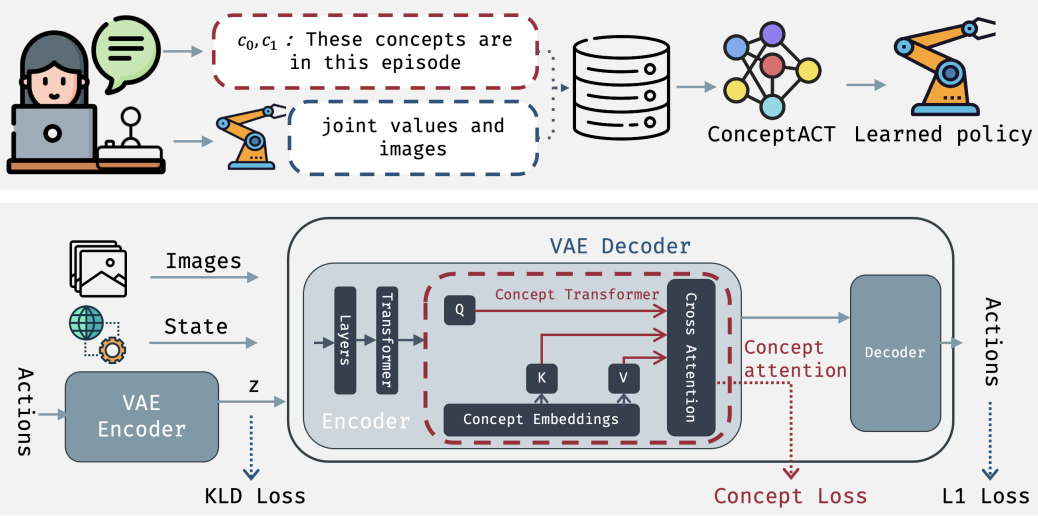


Figure 1: Overview of our approach: Top: We enhance the normal Imitation Learning approach by allowing the user to specify which concepts are in an episode. Bottom: We integrate these concepts by changing the ACT architecture to include a Concept Transformer, which aligns its attentions mechanism to match with the given concepts. This alignment cost can then be included in the total loss. Red indicates changes.

We address this limitation by introducing ConceptACT, an extension of the ACT architecture that integrates episode-level semantic concepts directly into the imitation learning process. Our approach enables human demonstrators to annotate episodes with high-level concepts (such as object colors, shapes, or spatial relationships) and uses these annotations as auxiliary supervision during training. By incorporating a Concept Transformer module into the standard ACT encoder, ConceptACT learns to attend to semantically meaningful concepts while predicting action sequences, creating stronger inductive biases that improve sample efficiency and task understanding. We evaluate ConceptACT on a robotic pick-and-place task with sorting constraints, where robots must manipulate objects of varying shapes and colors according to complex conditional rules.

The primary contributions of this work are threefold: (1) we demonstrate a systematic approach for integrating episode-level semantic concepts into transformer-based imitation learning, (2) we show that proper architectural integration of concepts through attention mechanisms provides superior learning compared to auxiliary prediction tasks, and (3) we provide empirical evidence that concept-guided learning improves sample efficiency in manipulation tasks requiring conditional reasoning about object properties.

2 Background

We formulate sequential decision-making problems within the framework of Markov Decision Processes (MDPs). An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1]$ the discount factor.

At each timestep t, an agent observes state $s_t \in \mathcal{S}$, executes action $a_t \in \mathcal{A}$, and transitions to a new state s_{t+1} according to the transition dynamics $\mathcal{P}(s_{t+1}|s_t, a_t)$. The agent's objective is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward.

In many practical applications, including robotic manipulation, states s_t comprise diverse sensory inputs such as proprioceptive measurements and visual observations, while actions a_t correspond to motor commands or target configurations.

2.1 Imitation Learning and Behavior Cloning

Imitation learning addresses scenarios where specifying an explicit reward function proves challenging or impractical, but expert demonstrations are readily available. This paradigm is particularly valuable in domains where task success is easier to demonstrate than to define formally.

Given a dataset of expert demonstrations $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^{N}$ consisting of state-action pairs collected from expert trajectories, the goal is to learn a policy π_θ parameterized by θ that mimics the expert's behavior. The underlying assumption is that the expert demonstrations are generated by an optimal or near-optimal policy π^*.

Behavior Cloning represents the most direct approach to imitation learning, formulating the problem as supervised learning. The policy parameters θ are optimized to minimize the discrepancy between predicted and demonstrated actions:

$$\mathcal{L}_{BC}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\ell(\pi_\theta(s), a)] \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes an appropriate loss function—typically L1 or L2 norm for continuous action spaces, or cross-entropy for discrete actions.

2.2 Action Chunking with Transformers (ACT)

Action Chunking with Transformers (ACT) [Zhao et al., 2023] represents a significant advancement in imitation learning for robotic manipulation, introducing two key architectural innovations that improve policy learning from human demonstrations. Rather than predicting single actions at each timestep, ACT predicts sequences of actions (chunks) and employs a variational autoencoder (VAE) framework with transformer encoder-decoder architecture. This approach has proven particularly effective for fine manipulation tasks requiring temporal consistency and precise control.

2.2.1 Action Chunking

Traditional behavior cloning learns a policy $\pi_\theta(a_t|s_t)$ that predicts the immediate next action. ACT instead learns a policy that predicts action sequences:

$$\pi_\theta(a_{t:t+k-1}|s_t) = \pi_\theta(a_t, a_{t+1}, \ldots, a_{t+k-1}|s_t) \tag{2}$$

This reduces the effective horizon of a task from T timesteps to $\lceil T/k \rceil$ decision points, mitigating error accumulation. Action chunking also helps model non-Markovian behavior in human demonstrations, such as natural pauses that single-step policies struggle to handle.

2.2.2 Variational Training

One key part of ACT is the usage of VAE-style training[1] [Kingma and Welling, 2014]. The VAE-Encoder has the goal of producing a suitable latent "style" variable $z \in \mathbb{R}^L$ which represents the characteristics of the current trajectory. To achieve this, the VAE-Encoder is trained on the whole sequence of joint variables as input (but not other inputs like images to reduce computational needs) with a KL loss.

The VAE-Encoder infers the posterior distribution:

$$q_\phi(z|s_t, a_{t:t+k-1}) = \mathcal{N}(\mu_\phi(s_t, a_{t:t+k-1}), \sigma^2_\phi(s_t, a_{t:t+k-1})) \tag{3}$$

This latent variable z is then used as input for the VAE-Decoder. At inference time, the latent variable z is set to zero, ensuring deterministic predictions.

The complete training objective combines reconstruction accuracy with latent regularization:

$$\mathcal{L}_{ACT} = \mathbb{E}_{q_\phi}[\ell(a_{t:t+k-1}, p_\theta(a_{t:t+k-1}|s_t, z))] + \beta \cdot D_{KL}[q_\phi(z|s_t, a_{t:t+k-1})\|\mathcal{N}(0, I)] \tag{4}$$

where $\ell(\cdot, \cdot)$ is typically the L1 loss for continuous actions.

[1]For writing clarity we refer to VAE Encoder/Decoder as parts of the larger VAE architecture. Encoder/Decoder then refer to the transformer components inside the VAE-Decoder.

2.2.3 Transformer Architecture

The VAE-Decoder consists of a classical transformer encoder-decoder architecture [Vaswani et al., 2017]. The encoder receives the latent variable z, current joint (and environmental) variables, and images as input. Images are usually embedded by pre-trained ImageNet encoders. The goal of the encoder is to produce a sequence of latent embeddings which are used by the decoder.

The decoder part then receives this encoder output and additionally uses fixed positional embeddings. In ACT, the model is trained to predict a whole chunk of actions via L1-Loss, in contrast to normal behavior cloning where only a single next action would be predicted.

For improved temporal consistency, ACT employs temporal ensembling during inference: overlapping action chunks are combined through exponentially weighted averaging of predictions for each timestep.

2.3 Concept Transformers

Traditional deep learning models operate on low-level features that often lack human-interpretable meaning. To address this limitation, concept-based methods aim to ground model decisions in high-level concepts that align with human understanding. The Concept Transformer [Rigotti et al., 2022] extends this paradigm by generalizing attention mechanisms from low-level input features to high-level interpretable concepts. The key idea is to modify the standard multi-headed attention mechanism to explicitly attend over user-defined concepts.

Instead of the standard query, key, and value projections all derived from the input, the Concept Transformer only projects the input into queries. Crucially, the keys and values are instead derived from a fixed set of learnable concept embeddings that are independent of the input.

Given input features $X \in \mathbb{R}^{P \times d}$ (where P is the number of patches or tokens) and a set of C learnable concept embeddings $\mathbf{c}_1, \ldots, \mathbf{c}_C \in \mathbb{R}^d$, the attention mechanism computes: $Q = XW_Q$, $K = CW_K$, and $V = CW_V$, where $C = [\mathbf{c}_1; \ldots; \mathbf{c}_C] \in \mathbb{R}^{C \times d}$ represents the concatenated concept embeddings, and $W_Q \in \mathbb{R}^{d \times d}$, $W_K \in \mathbb{R}^{d \times d}$, $W_V \in \mathbb{R}^{d \times d}$ are learned projection matrices. The attention weights are computed as:

$$\alpha_{pc} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)_{pc} \tag{5}$$

A crucial difference from standard attention is that these attention scores α_{pc} can be interpreted as importance weights over the given concepts. In the original work [Rigotti et al., 2022], these attention scores are aligned with ground-truth concept annotations through supervision using the Frobenius norm:

$$\mathcal{L}_{concept} = \|A - H\|_F^2 \tag{6}$$

where $A = [\alpha_{pc}] \in \mathbb{R}^{P \times C}$ represents the computed attention and H indicates which concepts should be attended to for each input. This supervision ensures that learned attention patterns respect domain knowledge, making them both plausible (convincing to humans) and faithful (reflective of the model's reasoning).

The total loss combines the primary task objective with concept supervision:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda \mathcal{L}_{concept} \tag{7}$$

where λ controls the relative importance of concept alignment.

3 Implementation

We extend the standard imitation learning setting to incorporate high-level semantic information. In addition to the demonstration dataset $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^N$ of state-action pairs, we assume access to concept annotations $\mathcal{C} = \{c_i\}_{i=1}^N$, where each c_i represents a binary vector indicating the presence of specific concepts in step i.

Formally, for a set of concept types $\mathcal{T} = \{T_1, T_2, \ldots, T_K\}$ (e.g., object colors, shapes), each concept annotation c_i is composed of $c_i = [c_i^{T_1}, c_i^{T_2}, \ldots, c_i^{T_K}]$, where $c_i^{T_j} \in \{0,1\}^{|T_j|}$ indicates which specific concept within type T_j is present in episode i. For instance, if an episode contains a red object, then $c_i(\text{color} = \text{red}) = 1$.

3.1 ConceptACT Architecture

To incorporate concept learning into ACT training, we modify the architecture by replacing the final layer of the transformer encoder (i.e., the encoder part of the VAE-Decoder) with a concept-aware layer. We implement two distinct approaches for concept integration.

3.1.1 Method 1: Prediction Head Approach

In this approach, we augment the standard ACT encoder with separate prediction heads for each concept type. The encoder output representation is passed through concept-specific networks that produce logits for each concept type. These predictions are trained using cross-entropy loss against the ground-truth concept annotations:

$$\mathcal{L}_{\text{concept}}^{\text{CE}} = \sum_{j=1}^{K} \mathbb{E}[\text{CrossEntropy}(\hat{c}^{T_j}, c^{T_j})] \tag{8}$$

where $\hat{c}^{T_j}$ are the predicted logits and c^{T_j} are the ground-truth labels for concept type T_j.

3.1.2 Method 2: Concept Transformer Integration

Alternatively, we replace the final encoder layer with a concept-aware transformer layer inspired by Concept Transformers [Rigotti et al., 2022]. This layer computes cross-attention between input features and learnable concept embeddings.

Given encoder input $X \in \mathbb{R}^{S \times d}$ and concept embeddings $E \in \mathbb{R}^{C \times d}$ (where $C = \sum_{j=1}^{K} |T_j|$), we compute standard attention: $Q = XW_Q$, $K = EW_K$, $V = EW_V$.

Crucially, we make two key modifications to the standard Concept Transformer. First, instead of using the Frobenius norm to align concept attention (which would be computationally identical to MSE), we employ a Binary Cross Entropy loss formulation for multi-label concept prediction.

Second, since our transformer uses multi-headed attention, we do not use simple averaging of concept attention scores across heads. Instead, we learn a dynamic weighting of attention heads through a small neural network:

$$w = \text{softmax}(\text{MLP}_{\text{weight}}(\bar{X})) \tag{9}$$

where $\bar{X} = \frac{1}{S}\sum_{s=1}^{S} X_s$ is the mean-pooled input representation, and $w \in \mathbb{R}^H$ represents the learned weights for H attention heads.

The final concept predictions are obtained by weighted aggregation:

$$\hat{c} = \sum_{h=1}^{H} w_h \cdot \text{mean}_S(A_h) \tag{10}$$

where A_h represents the attention scores from head h.

The complete ConceptACT training objective combines the standard ACT losses with concept prediction:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{ACT} + \lambda_{\text{concept}} \mathcal{L}_{\text{concept}} \tag{11}$$

where $\mathcal{L}_{ACT}$ includes both the action reconstruction and KL divergence terms as defined in Section 2.3.

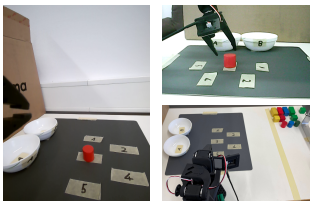


Figure 2: Left: Wrist Camera, Top: Scene Camera. Bottom: Experimental setup.

Method	Optimality Gap	95% CI
ACT	0.549	[0.48, 0.58]
ConceptACT - Heads	0.515	[0.45, 0.56]
ConceptACT - Transformer	0.317	[0.27, 0.40]

Table 1: Optimaltiy Gap (lower is better) for each variant. Due to the overlapping CI we can only claim significance of the Concept Transformer method.

For concept prediction, we implement Binary Cross Entropy loss to handle the multi-label nature of our concept annotations:

$$\mathcal{L}_{\text{concept}} = \mathbb{E}[\text{BCE}(\hat{c}, c)] \tag{12}$$

where c represents the concatenated ground-truth concept vector across all types.

After training is complete, the concept prediction components can be discarded during inference, allowing the policy to be used normally for action prediction. This ensures that concept learning aids training without affecting deployment efficiency.

4 Evaluation

While ConceptACT is generally applicable to any imitation learning domain, we evaluate our approach in a robotic manipulation setting for several practical reasons. First, robotic tasks naturally exhibit the complex state-action relationships where concept-based guidance can provide the most benefit. Second, the visual and proprioceptive nature of robotic observations allows for intuitive concept definitions (e.g., object properties, spatial relationships). Finally, the precision requirements of manipulation tasks create scenarios where improved sample efficiency from concept learning translates to meaningful performance gains.

4.1 Hardware Configuration

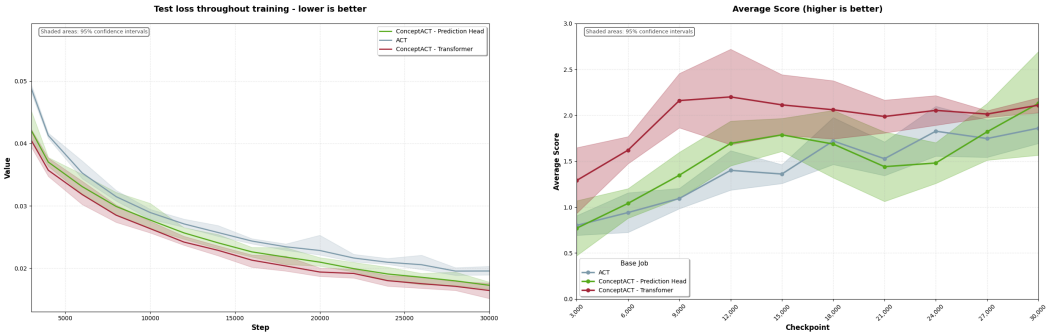
Our experimental setup employs a bilateral robotic system consisting of two identical SO-100 robotic arms. One arm serves as the **leader** for human demonstration and teleoperation, while the other functions as the **follower** for policy execution and evaluation. This leader-follower configuration enables intuitive data collection: human demonstrators directly manipulate the leader arm through kinesthetic teaching, with joint positions and trajectories recorded in real-time. The system is equipped with two cameras providing complementary viewpoints: a **gripper camera** attached to the follower arm's end-effector for detailed manipulation views, and a **scene camera** positioned to capture the entire workspace from a fixed overhead perspective. This dual-camera setup ensures comprehensive visual coverage of both fine-grained manipulation details and global scene context.

4.2 Task Description and Concept Design

The experimental task involves pick-and-place operations with an additional sorting constraint, requiring the robot to grasp objects of varying shapes and colors and sort them into two designated collection areas. The workspace contains objects with three distinct shapes (cube, cylinder, rectangular prism) in four colors (green, red, blue, yellow), though not all shape-color combinations are present. Objects are randomly placed at five different locations across the table surface. The sorting rule determines the target collection area based on both shape and color properties:

$$\text{Target} = \begin{cases} \text{Area A} & \text{if } (\text{shape} = \text{cube} \wedge \text{color} \in \{\text{red}, \text{green}\}) \\ & \text{or } (\text{shape} = \text{cylinder} \wedge \text{color} = \text{blue}) \\ \text{Area B} & \text{otherwise} \end{cases} \tag{13}$$

This task design creates a scenario where pure behavioral cloning of pick-and-place motions is insufficient—the policy must also learn the underlying sorting logic to achieve optimal performance.



(a) ACT loss on a testing dataset during training

(b) Robot evaluation performance

Figure 3: Training and evaluation results comparing ACT, ConceptACT with prediction heads, and ConceptACT with Concept Transformer. (a) Test loss on 20% holdout set throughout training. (Lower is better) (b) Real robot evaluation performance measured every 3,000 training steps on 10 test episodes. (Higher is better) - Each method was trained with 5 different random seeds. Bold lines represent means, shaded areas show variance across runs.

4.3 Data Collection and Concept Annotation

Human demonstrators performed the task using the leader arm, with episodes recorded at 50Hz including joint positions, camera feeds, and timestamped action sequences. Crucially, during each demonstration episode, we annotated the session with high-level concept information corresponding to the object being manipulated.

For each episode i, we recorded concept vectors c_i indicating shape concepts $c_i^{\text{shape}} \in \{0,1\}^3$ (one-hot encoding for cube, cylinder, rectangular prism), color concepts $c_i^{\text{color}} \in \{0,1\}^4$ (one-hot encoding for red, green, blue, yellow), and location concepts $c_i^{\text{location}} \in \{0,1\}^5$ (one-hot encoding for drop-off positions).

This concept annotation process occurred during data collection, with demonstrators explicitly identifying object properties at the beginning of each episode. We collected a total of 200 demonstration episodes with roughly balanced distributions across shape, color, and pickup location combinations. To evaluate generalization, we reserved specific shape-color-location combinations exclusively for testing, ensuring that policy evaluation includes both seen and unseen concept combinations.

4.4 Evaluation Metrics

Given the multi-faceted nature of the task, we employ a hierarchical scoring system that captures different levels of task completion. We assign a score of 0 if the pick attempt failed (object not successfully grasped), a score of 1 for successful pick but failed place attempt (object dropped or misplaced), a score of 2 for successful pick and place but incorrect sorting (object placed in wrong collection area), and a score of 3 for complete success (correct pick, place, and sorting according to the rule). We collect this metrics on 10 test-cases (which are not in training dataset) at multiple training checkpoints (every 3k steps), for multiple training runs (each with a different random seed). This allows us to showcase the difference in real-world learning progress between our baseline (ACT) and our addition.

4.5 Results

As shown in Figure 3, ConceptACT with Concept Transformer integration demonstrates accelerated learning, particularly during early training phases. This improvement is evident in both the test loss metrics (Figure 3a) and real robot evaluation performance (Figure 3b).

The performance gap between methods becomes less pronounced in later training phases, which we attribute to the relative simplicity of the current task given the available demonstration data. Our experimental design first validated task feasibility with standard ACT before evaluating our

extensions. Nevertheless, ConceptACT's faster convergence provides substantial benefits in scenarios where training time is critical or demonstration data is limited.

Notably, the prediction head variant of concept integration yields minimal improvement over standard ACT. This result demonstrates that merely incorporating additional concept information is insufficient—the integration method is crucial for realizing performance gains. Naive concept inclusion without proper architectural considerations provides little benefit.

During evaluation, we observed qualitatively different failure modes between methods. Standard ACT policies frequently dropped objects precisely at the boundary between collection areas, sometimes even balancing objects on the edge—a behavior indicating successful pick-and-place learning but failure to internalize the sorting rule. In contrast, ConceptACT policies never exhibited this boundary confusion, suggesting better acquisition of the underlying sorting logic encoded in the concept annotations.

To complement the temporal learning curves, we computed optimality gaps for each method and report means with 95% confidence intervals using bootstrapping following Agarwal et al. [2021]. Table 1 shows that while both concept-based methods improve the optimality gap, only the Concept Transformer integration achieves statistically significant improvement over the baseline ACT method.

5 Related Work

Our work integrates high-level semantic concepts into imitation learning to improve sample efficiency and interpretability. Here we review related approaches that incorporate auxiliary information, particularly concepts, into various learning paradigms.

[Hristov and Ramamoorthy, 2021] label demonstrated trajectories with high-level spatial concepts ("behind", "on top") and temporal concepts ("quickly"). Their method learns disentangled representations that separate task-relevant features from task-irrelevant variations. While they focus on trajectory-level semantic labeling similar to our episode-level concepts, their approach targets explainability through disentanglement rather than improving sample efficiency through auxiliary supervision.

[Cubek et al., 2015] employ conceptual spaces theory for learning from demonstration, using subspace clustering to identify relevant conceptual dimensions from sensory data. Their method automatically discovers conceptual representations that bridge low-level sensory input and high-level task understanding but relies on hard-coded high-level actions and leverages symbolic planning. Unlike their unsupervised concept discovery, ConceptACT uses human-provided concept annotations during training to guide the learning process explicitly.

[Stepputtis et al., 2020]) combine natural language, vision, and motion for abstract task representation in imitation learning. Their multi-modal approach learns policies conditioned on language instructions, enabling generalization across task variations. While both approaches integrate high-level semantic information, their language conditioning operates at inference time for task specification, whereas ConceptACT uses concept annotations during training to improve learning efficiency.

The idea of restricting the network structures to concepts has been quite successfully investigated in supervised learning. Besides Concept Transformer Rigotti et al. [2022], other notable approaches include Concept Bottleneck Models Koh et al. [2020] and Concept Whitening Chen et al. [2020]. Although most focus has been on interpretability, recent work demonstrates that concept-based auxiliary supervision can significantly improve learning performance. For instance, Fontana et al. [2024] show how auxiliary concept tasks enhance computer vision performance through multi-task learning, while Yao et al. [2023] demonstrate that semantic auxiliary tasks improve retrieval performance even with limited annotations. Similarly, Mahapatra et al. [2020] integrate semantic guidance into GAN learning, showing improved classification and segmentation performance. These works highlight that concept supervision provides not just interpretability but also serves as a powerful inductive bias that improves sample efficiency and generalization.

Our ConceptACT approach is distinguished by its integration of semantic concept learning directly into the action chunking transformer architecture for imitation learning. Unlike methods that discover concepts unsupervised or use them solely for interpretability, we leverage human-provided concept annotations as auxiliary supervision to improve sample efficiency. Furthermore, while most concept-

based approaches in RL focus on state abstraction or skill discovery, ConceptACT operates at the episode level, providing semantic guidance that helps the policy learn the underlying task structure more effectively. This positions our work at the intersection of interpretable machine learning and practical robotic imitation learning, addressing both the need for sample-efficient learning and human-understandable decision-making in robotic systems.

6 Discussion

Our experimental results demonstrate that incorporating high-level semantic concepts into imitation learning can significantly improve sample efficiency and task understanding.

The current evaluation is primarily constrained to a single robotic manipulation domain. While this experimental setting provides a sophisticated testbed with real-world complexities, broader evaluation across diverse environments—particularly non-robotic domains—would strengthen the generalizability claims of our approach. The pick-and-place with sorting task, though representative of many manipulation scenarios, represents only one class of problems where concept-guided learning might prove beneficial. The concept annotation process currently requires manual labeling during demonstration collection, which may limit scalability to more complex domains with richer concept spaces. Future work should investigate automated concept extraction methods or explore how to leverage existing knowledge bases to reduce the annotation burden on human demonstrators.

The superior performance of the Concept Transformer integration compared to simple prediction heads highlights the importance of architectural design in concept-based learning. This architectural difference suggests that effective concept integration requires more than simply adding concept prediction losses—it demands fundamental changes to how information flows through the network.

7 Conclusion

We introduced ConceptACT, an extension of the Action Chunking with Transformers architecture that incorporates episode-level semantic concepts directly into the imitation learning process. Our approach enables human demonstrators to provide high-level semantic annotations alongside behavioral demonstrations, creating auxiliary supervision that improves sample efficiency and task understanding.

Through controlled experiments on a robotic pick-and-place task with sorting constraints, we demonstrated that ConceptACT achieves faster convergence and better task performance compared to standard ACT, particularly during early training phases when demonstration data is limited. Crucially, we showed that the method of concept integration matters significantly—our Concept Transformer approach substantially outperforms naive prediction head integration, highlighting the importance of architectural design in concept-based learning.

Our findings suggest that the gap between human teaching and machine learning can be narrowed by leveraging the rich semantic knowledge that humans naturally provide during demonstration. This work opens promising directions for more sample-efficient and interpretable imitation learning systems that can benefit from human conceptual understanding while maintaining the flexibility and expressiveness of modern neural architectures. Future work should explore the scalability of this approach to more complex concept spaces, investigate automated concept discovery methods.

References

R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Z. Chen, Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

R. Cubek, W. Ertel, and G. Palm. High-level learning from demonstration with conceptual spaces and subspace clustering. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

M. Fontana, M. Spratling, and M. Shi. When multitask learning meets partial supervision: A computer vision review. *Proceedings of the IEEE*, 2024.

Y. Hristov and S. Ramamoorthy. Learning from demonstration with weakly supervised disentanglement. In *Ninth International Conference on Learning Representations 2021*, 2021.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

D. Mahapatra, B. Bozorgtabar, J.-P. Thiran, and L. Shao. Structure preserving stain normalization of histopathology images using self supervised semantic guidance. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*, pages 309–319. Springer, 2020.

M. Rigotti, C. Miksovic, I. Giurgiu, T. Gschwind, and P. Scotton. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations*, 2022.

S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. In *Advances in Neural Information Processing Systems*, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Y. Yao, Z. Zhang, K. Yang, H. Liang, Q. Yan, and Y. Xu. An auxiliary task boosted multi-task learning method for service account retrieval with limited human annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, 2023.

T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems XIX*, 2023.