PHILIPS

www.philips.com

Foundation Models for Sequential Decision Making

Large Pre-trained Causal Models

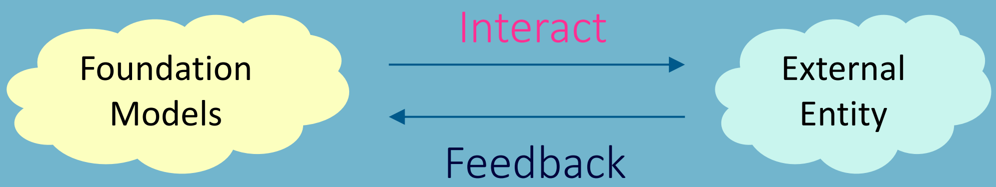
A Study Case of **Safety Critical Systems**

Shuyue Jia
M.Phil. Student
March 31st 2023

innovation + you

Foundation Models Roles

- **Generation Capability**
  Directly produce *action* or *state*
- **Representation Capability**
  Pre-trained learners of *states*, *actions*, *rewards*, and *transaction dynamics*



Interactive Decision Making

- Interact: Perform long-term reasoning, control, search, and planning
- Feedback: Solve tasks faster and generalize better

A Short Background of **Sequential Decision Making**

Task:

- *Learning from interactive experience (agent ↔ environment)*

Definition:

- *Markov Decision Process (MDP, Puterman, 1994)*

$$\mathcal{M} \coloneqq < S, A, R, ד, \mu, \gamma >$$

- S: state
- A: action (behavior)
- R: reward $R\colon\ S \times A \ \rightarrow\ \Delta(\mathbb{R})$
- ד: state transition function ד: $S \times A \ \rightarrow\ \Delta(S)$
- μ: initial state distribution $\mu \in \Delta(S)$
- γ: discount factor $\gamma \in [0, 1)$

$\boldsymbol{\pi}$: policy $\boldsymbol{\pi}\colon\ S \rightarrow \Delta(A)$

S_0: initial state $\mathrm{S}_0 \sim \mu$

Note: Expert Demonstrations

trajectory (episode):

state-action-reward tuples

$\tau_t \coloneqq (s_t, a_t, r_t)$

Goal and Method

Maximize the cumulative rewards of a policy through trial-and-error interactions with the env.

- *Reward*: total discounted sum of rewards $\mathrm{R}(\tau)$

$$\mathrm{R}(\tau) \coloneqq \sum_{\mathrm{t}=0}^{\mathrm{H}} \gamma^{\mathrm{t}} \mathrm{r}_{\mathrm{t}}$$

Maximizing $$\mathcal{J}(\pi) \coloneqq \mathbb{E}[\sum_{\mathrm{t}=0}^{\mathrm{H}} \gamma^{\mathrm{t}} \mathrm{r}_{\mathrm{t}} \,|\pi, \mathcal{M}]$$

- Imitation Learning *and* Behavior Cloning (BC)

  Train a policy π as close as π^*(expert demonstrations D_{RL})

  BC: directly *map state to action* via learning a policy π

$$L_{\mathrm{BC}}(\pi) \coloneqq \mathbb{E}_{(s,a)\sim D_{\mathrm{RL}}}\left[-\log(\pi(\mathrm{a}|\mathrm{s}))\right]$$

Methods Survey

$$\mathcal{T}(\pi) \coloneqq \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t r_t \,|\, \pi, \mathcal{M}\right]$$

Policy Gradient-based Methods

- *Estimate the gradient of* $\mathcal{T}(\pi)$ *w.r.t. the policy* π

Policy Gradient

$$\nabla_\theta \mathcal{T}(\pi_\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}}\left[\sum_{t=0}^{H} \gamma^t \boxed{\nabla_\theta \log \pi_\theta(a_t | s_t)} \hat{A}(s_t, a_t)\right]$$

Value-based Methods

- *Learn a optimal value function* $Q^*(s_t, a_t)$ *by satisfying Bellman Optimality Constraints*

$$\pi^*(\cdot\,|s_t) = \mathrm{argmax}_a Q^*(s_t, a)$$

Action-Value Function

$$Q^*(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim \tau(s_{t+1}|s_t, a_t)}\left[\max_{a_{t+1}} \boxed{Q^*(s_{t+1}, a_{t+1})}\right]$$

Actor-Critic Methods

- *First learn* $Q^\pi(s_t, a_t)$ *then learn a policy* π *by setting* $\hat{A}(s_t, a_t) = Q^\pi(s_t, a_t)$

Other Notes

Foundation Models for Decision Making

Modeling $p(\tau)$ from $\tau \sim D_{RL}$

Foundation Models:

- Self-supervised Learning on diverse data
- Task-specific Adaptation (Transfer Learning or Prompting)

Offline RL:

- Learn an algorithm from task specific RL dataset D_{RL}

Model-based RL: need to estimate *R* and $\daleth$ from dataset samples -> Learn a Model

Model-free RL: without R and $\daleth$ $\rightarrow$ learn policy and *R* via interactions

---

Goal: learn multimodal, multitask, and generalist interactive agents

Foundation Model Role 1: Generation Capability

Conditional Generative Models

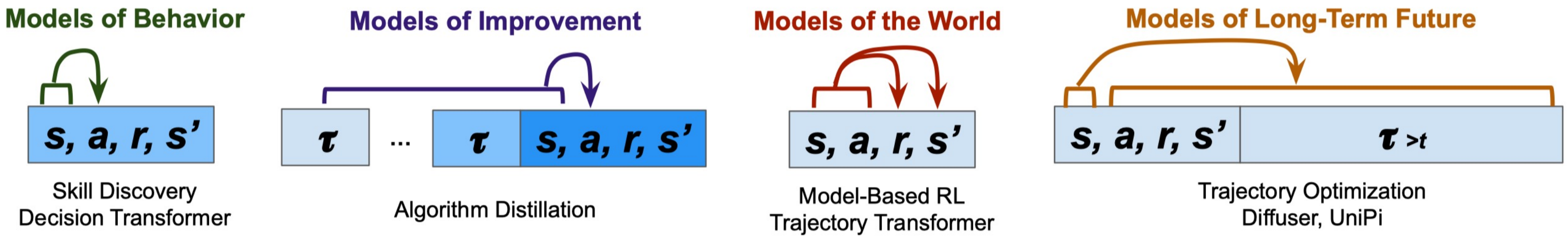


Fig. 3. Illustrations of how conditional generative models can model behaviors, improvements, environments, and long-term futures given a trajectory $\tau \sim \mathcal{D}_{\text{RL}}$. Dark blue indicates transitions with higher rewards. Models of behavior (Decision Transformers [Lee et al. 2022]) and self-improvement (Algorithm Distillation [Laskin et al. 2022]) require near-expert data. Models of the world (Trajectory Transformer [Janner et al. 2021]) and long-term future (UniPi [Du et al. 2023b]) generally require data with good coverage.

Foundation Model Role 1: Generation Capability

Conditional Generative Models

- *Definition: conditional* probability modeling of the trajectory distribution $p(\tau)$ from an interactive dataset $\tau \sim D_{\mathrm{RL}}$
- *Idea:* (1) Action (behaviors model)

  (2) Reward & State (environment dynamics, a.k.a. world model)
- *Difference:* factorization of $p(\tau)$ → conditional probabilities multiplication

$$p(x) = \prod_{l=1}^{\mathrm{L}} p(x_l | x_{<l}, z)$$

- *Latent Variable* z *:* represent different trajectory-level properties such as goals, skills, and dynamics constrains

Foundation Model Role 1: Generation Capability

Conditional Generative Models

- *Difference:* factorization of $p(\tau) \rightarrow$ conditional probabilities multiplication

$$p(x) = \prod_{l=1}^{L} p(x_l | x_{<l})$$

- *Summation*

$$L_{\mathrm{LM}}(p) \coloneqq \mathbb{E}_{x \sim D}\left[\sum_{l=1}^{L} -\log p(x_l | x_{<l})\right]$$

Foundation Model Role 1: Generation Capability

*Conditional Generative Models of **Behavior (Actions)** ← Pretraining*

$$L_{\mathrm{LM}}(\pi) := \mathbb{E}_{\tau\sim D_{\mathrm{RL}}}\left[\sum_{t=0}^{H} -\log \pi(a_t \mid \tau_{<t}, s_t)\right]$$

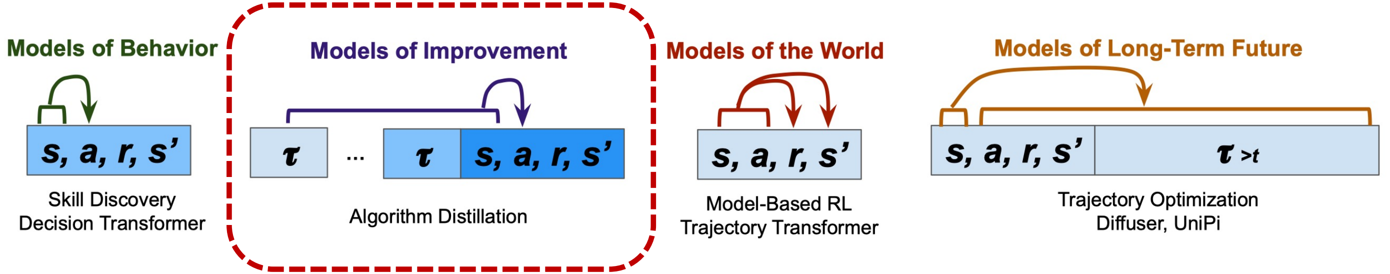


Fig. 3. Illustrations of how conditional generative models can model behaviors, improvements, environments, and long-term futures given a trajectory $\tau \sim \mathcal{D}_{\mathrm{RL}}$. Dark blue indicates transitions with higher rewards. Models of behavior (Decision Transformers [Lee et al. 2022]) and self-improvement (Algorithm Distillation [Laskin et al. 2022]) require near-expert data. Models of the world (Trajectory Transformer [Janner et al. 2021]) and long-term future (UniPi [Du et al. 2023b]) generally require data with good coverage.

Image Credit: Reference 1

Foundation Model Role 1: Generation Capability

Conditional Generative Models of ***Behavior (Actions)***

- *Policy that can depend on the history of interaction* $\pi(\mathrm{a_t}|\tau_{<\mathrm{t}}, \mathrm{s_t})$

  Encode *history* $(\tau_{<\mathrm{t}}, \mathrm{s_t})$ *and* ***decode*** *the next action* $\mathrm{a_t}$
- An additional conditioning variable ***z*** that *captures trajectory-level information*

$$L_{\mathrm{LM}}(\pi) := \mathbb{E}_{\tau\sim D_{\mathrm{RL}}}\left[\sum_{\mathrm{t}=0}^{\mathrm{H}} -\log \pi(\mathrm{a_t}|\tau_{<\mathrm{t}}, \mathrm{s_t}, \mathrm{z}(\tau))\right]$$

Others:

- Generalist Agents trained on massive behavior datasets
- Large-scale Online Learning

Foundation Model Role 1: Generation Capability

*Conditional Generative Models of **World (Environment Dynamics)***

- ***Idea**: Learn Transition Dynamics $\daleth$ and Reward Function R*

  from offline dataset $\tau \sim D_{\mathrm{RL}}$

  then improve policy π

- ***One-Step Prediction***

$$p(\tau) = \prod_{t=0}^{H} p(s_t, r_t, a_t | \tau_{<t}) = \prod_{t=0}^{H} \underbrace{\Gamma(s_t|\tau_{<t})}_{\text{Transition Dynamics}} \cdot \underbrace{\pi(a_t|\tau_{<t}, s_t)}_{\text{Behavior Policy}} \cdot \underbrace{\mathcal{R}(r_t|\tau_{<t}, s_t, a_t)}_{\text{Reward Function}}$$

- ***Long-Term Future***

$$p(\tau) = p(s_0, r_0, a_0, \ldots, s_H, r_H, a_H)$$

Foundation Model Role 2: Representation Capability

- *Plug-and-play style of knowledge compression and transfer*
- *Representation learning with task specifiers*
- *Learning representation for Sequential Decision Making*

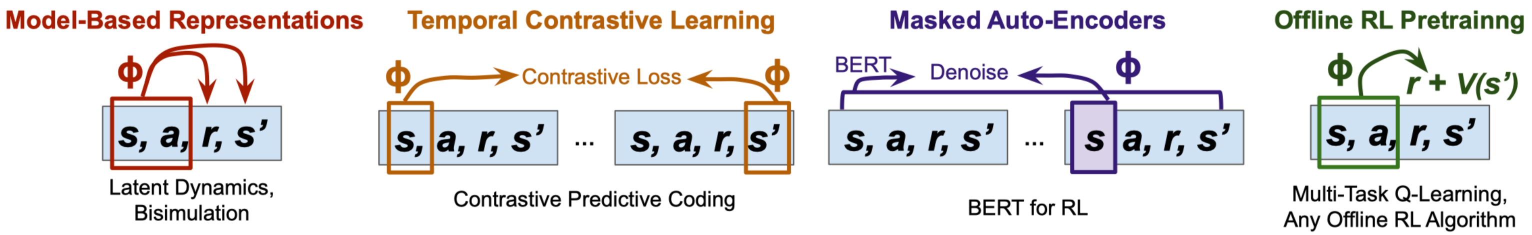


Fig. 4. Illustrations of different representation learning objectives such as model-based representations [Nachum and Yang 2021], temporal contrastive learning [Oord et al. 2018], masked autoencoders [Devlin et al. 2018], and offline RL [Kumar et al. 2022], on a trajectory $\tau \sim \mathcal{D}_{\mathrm{RL}}$ specifically devised for sequential decision making.

Image Credit: Reference 1

Foundation Model Role 2: Representation Capability

- *Model-based Representations*

Learning a latent state or action space of an env. by “clustering” states and actions that yield similar transition dynamics

$$\boldsymbol{\Gamma}(\mathrm{s}_{\mathbf{t+1}} | \boldsymbol{\tau}_{<t}, \emptyset(\boldsymbol{s}_t), \boldsymbol{a}_t)$$

$$\mathcal{R}(\boldsymbol{r}_{\mathrm{t}} | \boldsymbol{\tau}_{<t}, \emptyset(\boldsymbol{s}_t), \boldsymbol{a}_t)$$

$$\boldsymbol{\Gamma}(\emptyset(\mathrm{s}_{\mathbf{t+1}}) | \boldsymbol{\tau}_{<t}, \emptyset(\boldsymbol{s}_t), \boldsymbol{a}_t)$$

- *Temporal Contrastive Learning*
- *Masked Autoencoders*

Foundation Model Role 3: Agents and Environments

Agent

- *Learning from environment feedback produced by humans, tools, or the real world; Building new applications*
- *Example:* Optimize ChatGPT via RLHF
- *Example*: Generate API Calls (to invoke external tools and receive responses as feedback to support subsequent interaction)

Environment

- *Example:* Prompt ChatGPT

Foundation Models Significance

- **Generation Capability**
  Directly produce *action* or *state*
  Creativity

- **Representation Capability**
  Pre-trained learners of *states*, *actions*, *rewards*, and *transaction dynamics*
  Memorizing and Reasoning

Guang-Bin Huang
This is the reason why I called the intelligent revolution, exactly as Watt improved steam engine triggered Industrial Revolution

Like   Reply   1w

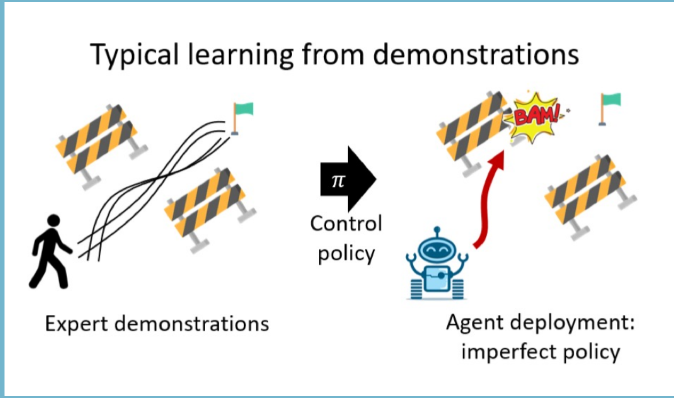
A Study Case: **Safety Critical System**

Paper: ConBaT: Control Barrier Transformer for Safe Policy Learning

Author: Yue Meng [1], Sai Vemprala [2], Rogerio Bonatti [2], Chuchu Fan [1], Ashish Kapoor [2]

Affiliation: MIT, Microsoft Research

Background and Goal



Background:

- Safety Requirement Scenario (*e.g.*, Safe Navigation)

Goal:

- Generate safe actions by learning a safe policy $\pi_{\text{safe}}: S \rightarrow A$

Image Credit: Reference 2

Previous Method and Proposed Method

Previous:

- Expert Demonstrations **with optimized safety constrains**
- **Cons**: unable to explicitly avoid unsafe actions; without unsafe behaviors

Motivation:

- Learn from safe and **unsafe demonstrations**
- Learn a safety critic on top of the control policy

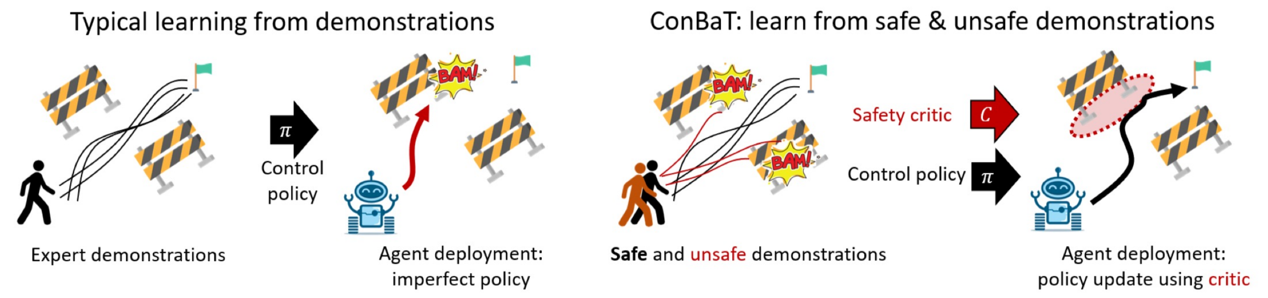


Figure 1: (Left) An agent trained to imitate expert demonstrations may just focus on the end result of the task without explicit notions of safety. (Right) Our proposed method ConBaT learns a safety critic on top of the control policy and uses the critic's control barrier to actively optimize the policy for safe actions.

Image Credit: Reference 2

Base Architecture: Perception-Action Causal Transformer (PACT)

Observation:

- Partially observable Markov decision process

  State-action tuples $\tau_t \coloneqq (s_t, a_t)$ and $t \in [0, T]$

Method – First Stage:

- State-action pairs from expert demonstrations to autoregressively train both a world model and a policy network, using imitation learning for its training objectives.

Image Credit: Reference 3

Base Architectu… …ormer (PACT)

Observation:

- Partially observa…

  State-action tup…

Method – First Stag…

- State-action pair… …train both a world model and a poli… …g objectives.

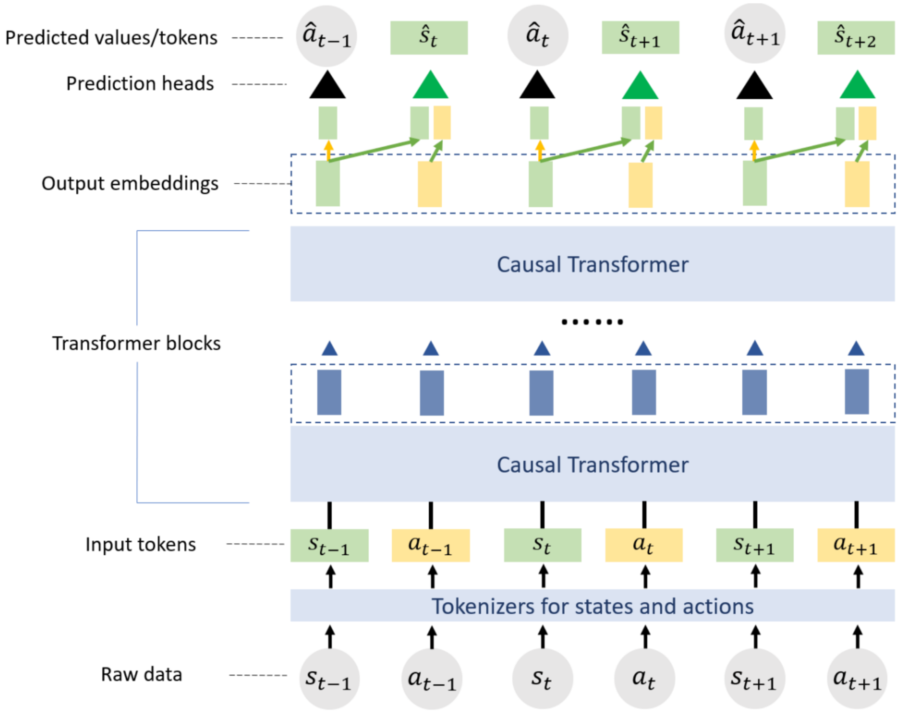


Fig. 2: Perception-Action Causal Transformer (PACT) architecture. $\hat{a}$ and $\hat{s}$ are autoregressively predicted actions and states. The tokenizer does not share information across data, and applies operations individually on raw data inputs. The black and green arrows represent predictions heads for actions and future state tokens respectively.

Image Credit: Reference 3

Perception-Action Causal Transformer (PACT)

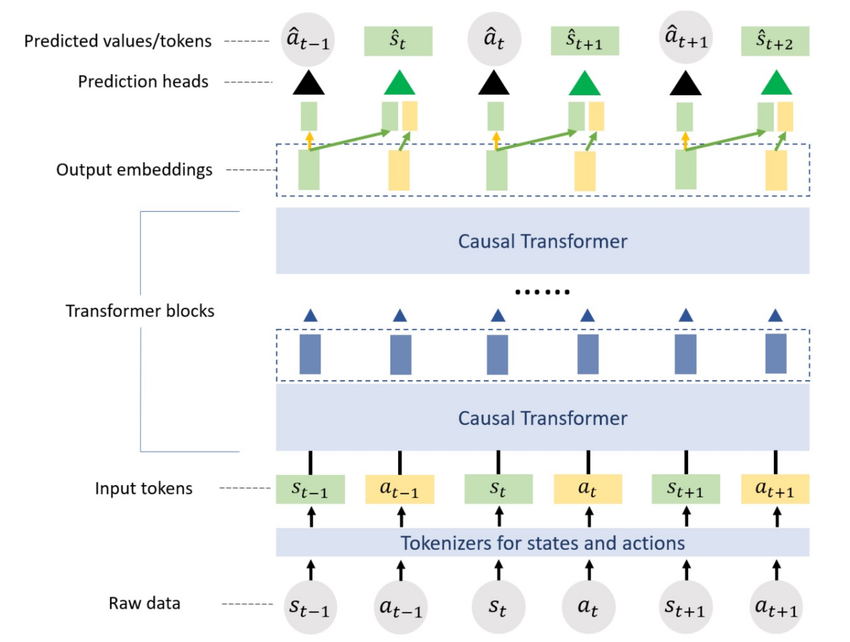


Fig. 2: Perception-Action Causal Transformer (PACT) architecture. $\hat{a}$ and $\hat{s}$ are autoregressively predicted actions and states. The tokenizer does not share information across data, and applies operations individually on raw data inputs. The black and green arrows represent predictions heads for actions and future state tokens respectively.

- *Tokenizer:*
  raw observation $\mathrm{s_t}$ and action $\mathrm{a_t}$ data
  $\rightarrow$
  compact tokens: $\mathrm{s'_t, a'_t} \in \mathbb{R}^{\mathrm{d}}$
  $$T_{\mathrm{s}}(\mathrm{s_t}) \rightarrow \mathrm{s'_t}$$
  $$T_{\mathrm{a}}(\mathrm{a_t}) \rightarrow \mathrm{a'_t}$$
- *Causal Transformer:*
  $$X(\mathrm{s'_0, a'_0}, \dots, \mathrm{s'}_T, \mathrm{a'}_T) \rightarrow (\mathrm{s}_0^+, \mathrm{a}_0^+, \dots, \mathrm{s}_T^+, \mathrm{a}_T^+)$$
- *Policy model:*
  $$\pi(\mathrm{s_t^+}) \rightarrow \mathrm{\hat{a}_t}$$
- *World model:*
  $$\emptyset(\mathrm{s_t^+, a_t^+}) \rightarrow \mathrm{s'_{t+1}}$$

Image Credit: Reference 3

This Work

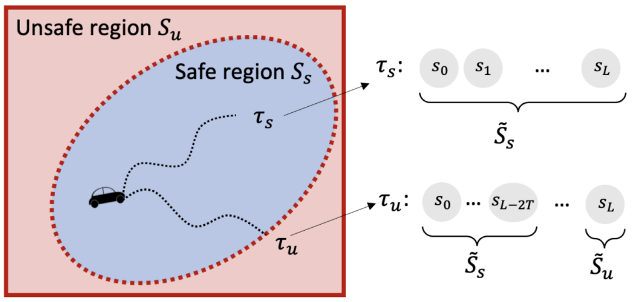


Fig. 2: Definitions of safe and unsafe sets. In safe demonstrations τ_s all state embeddings are labeled as safe. In contrast, in unsafe trajectories τ_u, only the first $(L - 2T)$ embeddings are assumed to be safe, where T is the Transformer context length, and only the last embedding is labeled as unsafe.

Observation:

- Two sets of trajectories in this work:

  (1) $\tau \in \sum_s$ → obey the desired safety constraints at all time steps

  (2) $\tau \in \sum_u$ → lead to an **unsafe terminal state**

Objective:

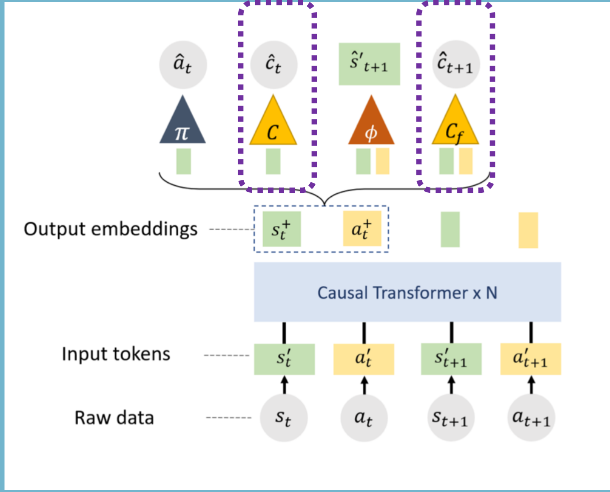
- Mimic the action distribution from good demonstrations $\sum_s$ (S_s)
- Avoiding sequences of actions that lead to the unsafe terminal states of $\sum_u$ (S_u)

Innovation – Control Barrier Critic

Two trainable critic modules:

→ ***predict safety scores*** *for the current and future expected states*

- $C: s_t^+ \rightarrow \hat{c}_t$
- $C_f: (s_t^+, a_t^+) \rightarrow \hat{c}_{t+1}$



Control Barrier Function (CBF):

- $\forall s \in S_s \rightarrow h(s) \geq 0$
- $\forall s \in S_u = \frac{S}{S_s} \rightarrow h(s) < 0$

$$\dot{h}(s) = \partial h(s)/\partial s \cdot f(s, \pi^*(s)) \geq -\alpha h(s) \ \text{ with } \ \alpha > 0$$

Image Credit: Reference 2

Training Critic Loss

Training the CBC involves three loss terms. First, we employ a classification loss $\mathcal{L}_c$ to enable the CBC to learn the safe set boundary:

$$\mathcal{L}_c = \mathop{\mathbb{E}}_{s_t^+ \sim \tilde{S}_s^+}\left[\sigma_+\left(\gamma - C(s_t^+)\right)\right] + \mathop{\mathbb{E}}_{s_t^+ \sim \tilde{S}_u^+}\left[\sigma_+\left(\gamma + C(s_t^+)\right)\right] \tag{4}$$

where $\sigma_+(x) = \max(x, 0)$ and γ is a margin factor that ensures numerical stability in training. The second loss enforces smoothness on the CBC values over time:

$$\mathcal{L}_s = \mathop{\mathbb{E}}_{s_t^+ \sim \tilde{S}^+}\left[\sigma_+\left((1-\alpha)C(s_t^+) - C(s_{t+1}^+)\right)\right] \tag{5}$$

where α controls the local decay rate. Note that this loss is asymmetrical as it only penalizes fast score decays but permits instantaneous increases, as a fast-improving safety level does not pose a problem. The final loss ensures consistency between the predictions of both critics C and C_f:

$$\mathcal{L}_f = \mathop{\mathbb{E}}_{s_t^+ \sim \tilde{S}^+}\left[\left|C_f(s_t^+, a_t^+) - C(s_{t+1}^+)\right|\right] \tag{6}$$

Theoretically, one could use a single critic C coupled with a world model ϕ to generate $\phi(s^+, a^+) \rightarrow \hat{s}'_{t+1}$ and then estimate future CBC score as $C(\hat{s}'_{t+1})$. We found it empirically helpful to use a separate critic head C_f to predict future CBC scores directly from the output embeddings, as it facilitates the action optimization process described in Section 2.2.2. The total training loss is $\mathcal{L}_{CB} = \lambda_c\mathcal{L}_c + \lambda_s\mathcal{L}_s + \lambda_f\mathcal{L}_f$, with relative weights λ.

Image Credit: Reference 2

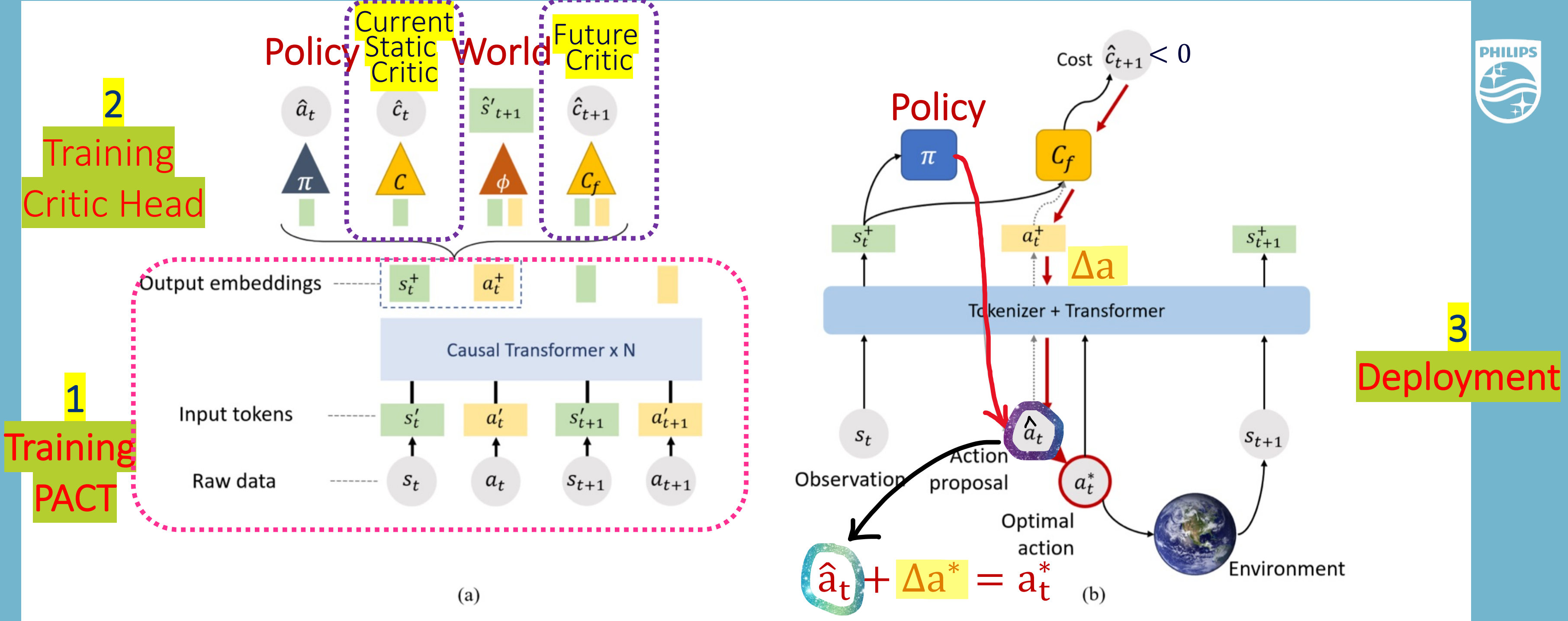
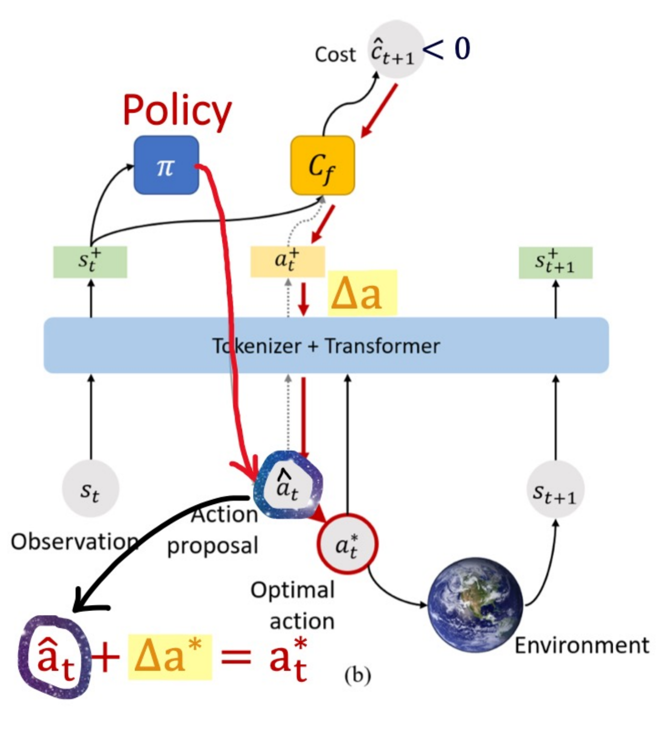


Figure 2: (a) The ConBaT architecture - a causal Transformer operates on state and action tokens (s', a') to produce embeddings (s^+, a^+). A policy head π computes actions given state embeddings, and a current state critic C computes a safety score. Both state and action embeddings are processed by a world model ϕ to compute the future state token, and by the future critic C_f to produce a future safety score. (b) The deployment process for ConBaT involves a feedback loop. The future critic evaluates action proposals from the policy head to check safety of resultant states. The red arrows show the flow of gradients that allow optimizing for the safe action that results in a desired cost characteristic. The optimal action a^* is used as the final command.

Key Point – Optimize Actions to Improve Safety



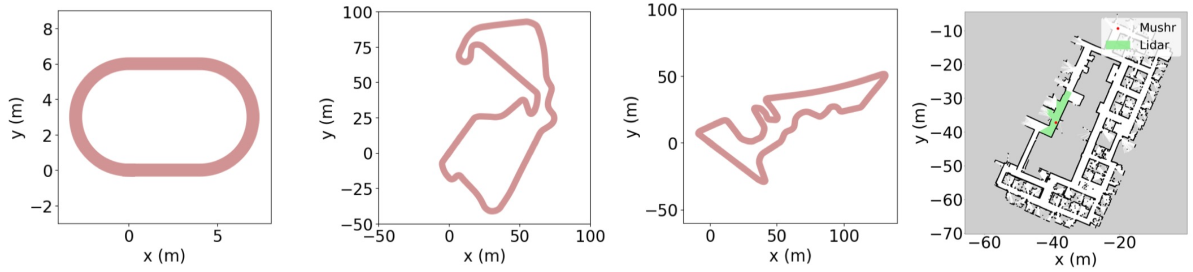
If $\hat{C}_{f,t+a} < 0$:

$$a_t^* = \hat{a}_t + \Delta a^*$$

Δa^*: gradient w.r.t. a

$$\Delta a^* = \text{argmin}_{\Delta a} \lambda \left|\left| \text{Cost}(\hat{C}_{t+1}, \text{unsafe label}) \right|\right| + \max(-C_f(s_t^+, a_t^+ + \Delta a), 0)$$

Databases (Simulated Environment)



(a) F1/10 (playground) (b) F1/10 (Silverstone) (c) F1/10 (Austin) (d) MuSHR environment

Figure 3: Simulation environment visualization.

F1/10 race car

- 2D Racing Tracks (Playground, Silverstone, and Austin)
- Observation: distance and angle; Action: steering angle

MuSHR car

- Observation: 2D LiDAR scan; Action: steering angle

Image Credit: Reference 2

Evaluation Metrics

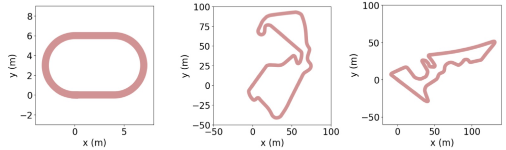
(1) Collision Rate

- The percentage of trajectories in the test set that end in a crash within the cut-off time horizon

(2) Average Trajectory Length (ATL)

- The average length of deployment trajectories, expressed in number of time steps before crashing or time-out if no crash occurs.

Databases (Simulated Environment)



(a) F1/10 (playground) (b) F1/10 (Silverstone) (c) F1/10 (Austin)

Figure 3: Simulation environment visualization.

	PACT	PACT-FT	ConBaT
Playground	100	-	**0.0**
Silverstone	100	96.88	**0.0**
Austin	100	100	**61.7**

(a) Collision Rate (%) - lower is better

	PACT	PACT-FT	ConBaT
Playground	175.45	-	**1000**
Silverstone	61.57	439.28	**1000**
Austin	57.11	165.12	**678.14**

(b) Avg. Trajectory Length - higher better

Table 1: Comparison of PACT and ConBaT for the F1/10 task. ConBaT outperforms PACT

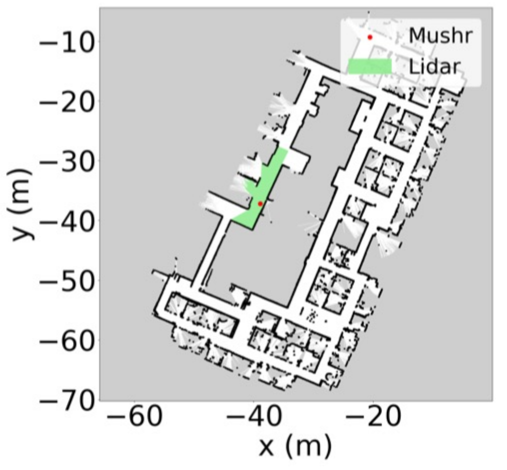
F1/10 race car – Playground

Train: 1K demonstrations, each 100 timesteps long

Test: 128 trajectories for a maximum of 1000 timesteps

Image Credit: Reference 2

Databases (Simulated Environment)



(d) MuSHR environment

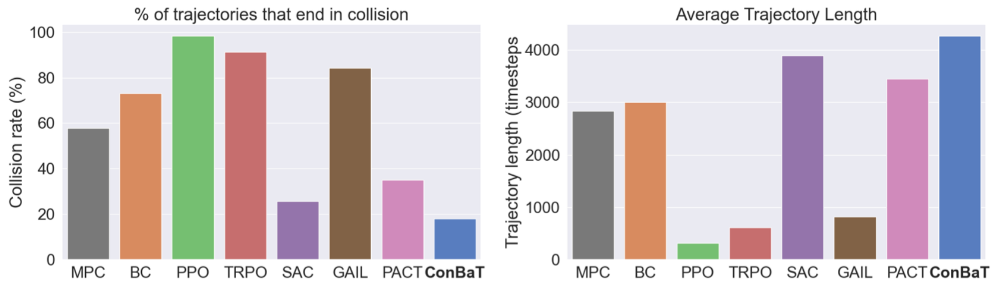


Figure 5: ConBaT outperforms classical MPC and several learning-based methods on safe navigation in the 2D MuSHR car domain.

MuSHR car

Train: 10K trajectories

Test: 128 trajectories for a maximum of 5000 timesteps

Image Credit: Reference 2

Potential Improvement

- (State, Action) ↔ Safe or Unsafe

  In the real-world scenario, it should be \`Fuzzy\` with a probability.

  Can we integrate or consider \`Fuzzy Control\` into this system?

- Reward Design

  Non-collision rate can be regarded as a reward, right?

  Can we design a new framework also with the consideration of maximizing the reward?

References:

(1) Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., & Schuurmans, D. (2023).

Foundation Models for Decision Making: Problems, Methods, and Opportunities.

arXiv preprint arXiv: 2303.04129.

(2) Meng, Y., Vemprala, S., Bonatti, R., Fan, C., & Kapoor, A. (2023).

ConBaT: Control Barrier Transformer for Safe Policy Learning.

arXiv preprint arXiv:2303.04212.

(3) Bonatti, R., Vemprala, S., Ma, S., Frujeri, F., Chen, S., & Kapoor, A. (2022).

PACT: Perception-Action Causal Transformer for Autoregressive Robotics Pre-Training.

arXiv preprint arXiv:2209.11133.