Task Programming: Learning Data Efficient Behavior Representations
任務編程：學習數據高效行為表徵

Jennifer J. Sun, Ann Kennedy, Eric Zhan, David J. Anderson, Yisong Yue, Pietro Perona

Caltech
Northwestern University

CVPR 2021 Best Paper Candidate
https://arxiv.org/abs/2011.13917
Comments: To appear in as an Oral in CVPR 2021.
Code & Project Website: https://sites.google.com/view/task-programming

Abstract 摘要

Specialized domain knowledge is often necessary to accurately annotate training sets for in-depth analysis, but can be burdensome and time-consuming to acquire from domain experts.
專業領域知識通常是準確註釋訓練集以進行深入分析所必需的，但從領域專家那裡獲取可能既繁瑣又耗時。

This issue arises prominently in automated behavior analysis, in which agent movements or actions of interest are detected from video tracking data.
這個問題在自動化行為分析中尤為突出，其中從視頻跟踪數據中檢測到代理移動或感興趣的動作。

To reduce annotation effort, we present TREBA: a method to learn annotation-sample efficient trajectory embedding for behavior analysis, based on multi-task self-supervised learning.
為了減少註釋工作，我們提出了 TREBA：一種基於多任務自監督學習學習註釋樣本有效軌跡嵌入以進行行為分析的方法。

The tasks in our method can be efficiently engineered by domain experts through a process we call "task programming", which uses programs to explicitly encode structured knowledge from domain experts.
我們方法中的任務可以由領域專家通過我們稱為"任務編程"的過程有效地設計，該過程使用程序對領域專家的結構化知識進行顯式編碼。

Total domain expert effort can be reduced by exchanging data annotation time for the construction of a small number of programmed tasks.
通過為構建少量編程任務交換數據註釋時間，可以減少領域專家的總工作量。

We evaluate this trade-off using data from behavioral neuroscience, in which specialized domain knowledge is used to identify behaviors.

我們使用來自行為神經科學的數據來評估這種權衡，其中使用專門的領域知識來識別行為。

We present experimental results in three datasets across two domains: mice and fruit flies.
我們在兩個領域的三個數據集中展示了實驗結果：小鼠和果蠅。

Using embeddings from TREBA, we reduce annotation burden by up to a factor of 10 without compromising accuracy compared to state-of-the-art features.
使用來自 TREBA 的嵌入，與最先進的特徵相比，我們在不影響準確性的情況下將註釋負擔減少了 10 倍。

Our results thus suggest that task programming and self-supervision can be an effective way to reduce annotation effort for domain experts.
因此，我們的結果表明，任務編程和自我監督可以成為減少領域專家註釋工作的有效方法。

1. Introduction 前言

Behavioral analysis of one or more agents is a core element in diverse fields of research, including biology [36, 26], autonomous driving [6, 39], sports analytics [42, 43], and video games [20, 3].
一個或多個代理的行為分析是不同研究領域的核心要素，包括生物學 [36, 26]、自動駕駛 [6, 39]、體育分析 [42, 43] 和視頻遊戲 [20, 3]。

In a typical experimental workflow, the location and pose of agents is first extracted from each frame of a behavior video, and then labels for experimenter-defined behaviors of interest are applied on a frame-by-frame basis based on the pose and movements of the agents.
在典型的實驗工作流程中，首先從行為視頻的每一幀中提取代理的位置和姿勢，然後根據行為視頻的姿勢和運動逐幀應用實驗者定義的感興趣行為的標籤。 代理商。

In addition to reducing human effort, automated quantification of behavior can lead to more objective, precise, and scalable measurements compared to manual annotation [1, 10].
除了減少人力，與手動註釋相比，行為的自動量化還可以帶來更客觀、更精確和可擴展的測量 [1, 10]。

However, training behavior detection models can be data intensive and manual behavior annotation often requires specialized domain knowledge and high-frequency temporal labels.
然而，訓練行為檢測模型可能是數據密集型的，手動行為註釋通常需要專門的領域知識和高頻時間標籤。

As a result, this process of generating training datasets is time-consuming and effort-intensive for experts.
因此，生成訓練數據集的過程對於專家來說既費時又費力。

Therefore, methods to reduce annotation effort by domain experts are needed to accelerate behavioral

studies.
因此，需要減少領域專家註釋工作的方法來加速行為研究。

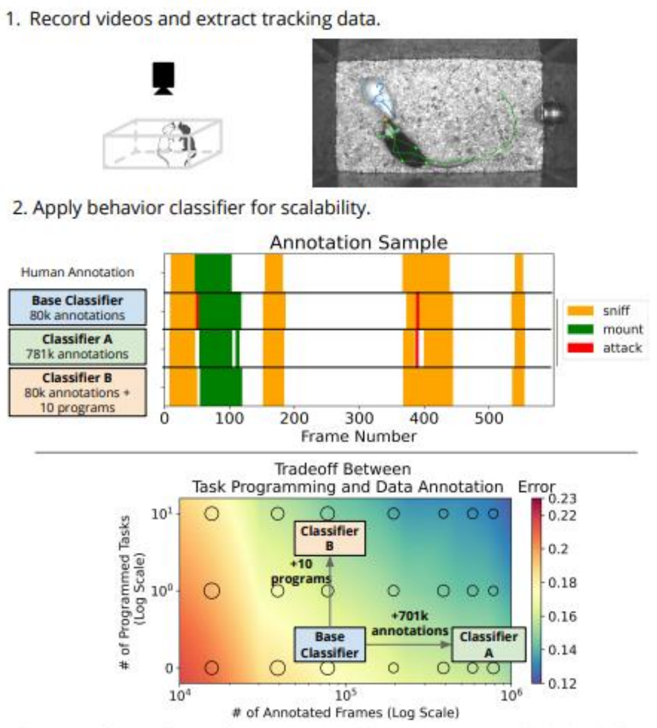


Figure 1. **Overview of our approach.** *Part 1:* A typical behavior study starts with extraction of tracking data from videos. We show 7 keypoints for each mouse, and draw the trajectory of the nose keypoint. *Part 2:* Domain experts can either do data annotation (Classifier A) or task programming (Classifier B) to reduce classifier error. The middle panel shows annotated frames at 30Hz. Colors in the bottom plot represent interpolated performance based on classifier error at the circular markers (full results in Section 4.3). The size of the marker represents the error variance.

Figure 1. Overview of our approach.
圖 1. 我們的方法概述。

Part 1:
第 1 部分：

A typical behavior study starts with extraction of tracking data from videos.
典型的行為研究始於從視頻中提取跟踪數據。

We show 7 keypoints for each mouse, and draw the trajectory of the nose keypoint.
我們為每隻鼠標顯示 7 個關鍵點，並繪製鼻子關鍵點的軌跡。

Part 2:
第 2 部分：

Domain experts can either do data annotation (Classifier A) or task programming (Classifier B) to reduce

classifier error.
領域專家可以進行數據註釋（分類器 A）或任務編程（分類器 B）以減少分類器錯誤。

The middle panel shows annotated frames at 30Hz.
中間面板以 30Hz 顯示帶註釋的幀。

Colors in the bottom plot represent interpolated performance based on classifier error at the circular markers (full results in Section 4.3).
底部圖中的顏色表示基於圓形標記處的分類器誤差的插值性能（第 4.3 節中的完整結果）。

The size of the marker represents the error variance.
標記的大小代表誤差方差。

We study alternative ways for domain experts to improve classifier accuracy beyond simply increasing the sheer volume of annotations.
我們研究領域專家提高分類器準確性的替代方法，而不僅僅是簡單地增加註釋的數量。

In particular, we propose a framework that unifies:
特別是，我們提出了一個統一的框架：

(1) self-supervised representation learning, and
(1) 自監督表徵學習，以及

(2) encoding explicit structured knowledge on trajectory data using expert-defined programs.
(2) 使用專家定義的程序對軌跡數據的顯式結構化知識進行編碼。

Domain experts can construct these programs efficiently because keypoint trajectories in each frame are typically low dimensional, and experts can already hand-design effective features for trajectory data [36, 28].
領域專家可以有效地構建這些程序，因為每幀中的關鍵點軌跡通常是低維的，並且專家已經可以為軌跡數據手動設計有效的特徵 [36, 28]。

To best leverage this structured expert knowledge, we develop a framework to learn trajectory representations based on multi-task self-supervised learning, which has not been well-explored for trajectory data.
為了最好地利用這種結構化的專家知識，我們開發了一個框架來學習基於多任務自監督學習的軌跡表示，該框架尚未對軌跡數據進行充分探索。

Our Approach. 我們的方法。

Our framework, Trajectory Embedding for Behavior Analysis (TREBA), learns trajectory representations through trajectory generation alongside a set of decoder tasks based on expert-engineered programs.

我們的框架，用於行為分析的軌跡嵌入 (TREBA)，通過軌跡生成以及一組基於專家設計的程序的解碼器任務來學習軌跡表示。

These programs are created by domain experts through a process we call task programming, inspired by the data programming paradigm [33].
這些程序是由領域專家通過我們稱為任務編程的過程創建的，受數據編程範式的啟發[33]。

Task programming is a process by which domain experts identify trajectory attributes relevant to the behaviors of interest under study, write programs, and apply those programs to inform representation learning (Section 3.2).
任務編程是領域專家識別與研究中感興趣的行為相關的軌跡屬性、編寫程序並將這些程序應用於表示學習的過程（第 3.2 節）。

This flexibility in decoder tasks allows our framework to be applicable to a variety of agents and behaviors studied across diverse fields of research.
解碼器任務的這種靈活性使我們的框架適用於跨不同研究領域研究的各種代理和行為。

Expert Effort Tradeoffs. 專家努力權衡。

Since task programming will typically require a domain expert's time, we study the tradeoff between doing task programming and data annotation.
由於任務編程通常需要領域專家的時間，因此我們研究了進行任務編程和數據註釋之間的權衡。

We compare behavior classification performance with different amounts of annotated training data and programmed tasks.
我們將行為分類性能與不同數量的帶註釋的訓練數據和編程任務進行比較。

For example, for the domain illustrated in Figure 1, domain experts can reduce error by 13% relative to the base classifier by annotating 701k additional frames, or they can reduce error by 16% by learning a representation using 10 programmed tasks in our framework.
例如，對於圖 1 中所示的領域，領域專家可以通過註釋 701k 額外幀，相對於基本分類器將錯誤減少 13%，或者通過在我們的框架中使用 10 個編程任務學習表示，他們可以將錯誤減少 16%。

Our approach allows experts to trade a large number of annotations for a small number of programmed tasks.
我們的方法允許專家用大量的註釋來交換少量的編程任務。

We study our approach across two domains in behavioral neuroscience, namely mouse and fly behavior.
我們在行為神經科學的兩個領域研究我們的方法，即小鼠和蒼蠅行為。

We chose this setting because it requires specialized domain knowledge for data annotation, and data efficiency is important for domain experts.

我們選擇這個設置是因為它需要專門的領域知識來進行數據標註，而數據效率對於領域專家來說很重要。

Furthermore, decoder tasks in our framework can be efficiently programmed by experts based on simple functions describing trajectory attributes for identifying behaviors of interest.
此外，我們框架中的解碼器任務可以由專家基於描述軌跡屬性的簡單函數有效地編程，以識別感興趣的行為。

For example, for mouse social behaviors such as attack [36], important behavior attributes include the speed of each mouse and distance between mice.
例如，對於攻擊等鼠標社交行為[36]，重要的行為屬性包括每隻鼠標的速度和鼠標之間的距離。

The corresponding task could then be to decode these attributes from the learned representations.
相應的任務可以是從學習到的表徵中解碼這些屬性。

Our contributions are: 我們的貢獻是：

・ We introduce task programming as an efficient way for domain experts to reduce annotation effort and encode structural knowledge.
・ 我們引入任務編程作為領域專家減少註釋工作和編碼結構知識的有效方式。

We develop a novel method to learn an annotation-sample efficient trajectory representation using self-supervision and programmatic supervision.
我們開發了一種使用自我監督和程序監督學習註釋樣本有效軌跡表示的新方法。

・ We study the effect of task programming, data annotation, and different decoder losses on behavior classifier performance.
・ 我們研究了任務編程、數據註釋和不同解碼器損失對行為分類器性能的影響。

・ We demonstrate these representations on three datasets in two domains, showing that our method can lead to a 10annotation reduction for mice, and 2for flies.
・ 我們在兩個域中的三個數據集上演示了這些表示，表明我們的方法可以導致小鼠減少 10 個註釋，蒼蠅減少 2 個註釋。

2. Related Work 相關工作

Behavior Modeling. 行為建模。

Behavior modeling using trajectory data is studied across a variety of fields [26, 6, 39, 42, 20, 3].
使用軌跡數據的行為建模在多個領域進行了研究 [26, 6, 39, 42, 20, 3]。

In particular, there is an increasing effort to automatically detect and classify behavior from trajectory data [23, 1, 14, 27, 13, 36].
特別是，越來越多的努力從軌跡數據中自動檢測和分類行為 [23, 1, 14, 27, 13, 36]。

Our experiments are based on behavior classification datasets from behavioral neuroscience [15, 4, 36], a field where specialized domain knowledge is important for identifying behaviors of interest.
我們的實驗基於來自行為神經科學 [15, 4, 36] 的行為分類數據集，該領域的專業領域知識對於識別感興趣的行為很重要。

The behavior analysis pipeline generally consists of the following steps:
行為分析管道一般包括以下步驟：

(1) tracking the pose of agents, (2) computing pose-based features, and (3) training behavior classifiers [4, 21, 36, 28].
(1) 跟踪代理的姿態，(2) 計算基於姿態的特徵，以及 (3) 訓練行為分類器 [4, 21, 36, 28]。

To address step 1, there are many existing pose estimation models [15, 27, 18, 36].
為了解決步驟 1，有許多現有的姿勢估計模型 [15、27、18、36]。

In our work, we leverage two existing pose models, [36] for mice and [15] for flies, to produce trajectory data.
在我們的工作中，我們利用兩個現有的姿勢模型，[36] 用於小鼠，[15] 用於蒼蠅，來生成軌跡數據。

In steps 2 and 3 of the typical behavior analysis pipeline, hand-designed trajectory features are computed from the animals' pose, and classifiers are trained to predict behaviors of interest in a fully supervised fashion [4, 21, 15, 36].
在典型行為分析流程的第 2 步和第 3 步中，根據動物的姿勢計算手工設計的軌跡特徵，並訓練分類器以完全監督的方式預測感興趣的行為 [4, 21, 15, 36]。

Training fully supervised behavior classifiers requires time-consuming annotations by domain experts [1].
訓練完全監督的行為分類器需要領域專家進行耗時的註釋 [1]。

Instead, our proposed approach enables domain experts to trade time-consuming annotation work for task programming with representation learning.
相反，我們提出的方法使領域專家能夠用表示學習來交換任務編程的耗時註釋工作。

Another group of work uses unsupervised methods to discover new motifs and behaviors [22, 41, 2, 26, 5].
另一組工作使用無監督方法來發現新的主題和行為 [22, 41, 2, 26, 5]。

Our work focuses on the more common case where domain experts already know what types of actions they would like to study in an experiment.
我們的工作側重於更常見的情況，即領域專家已經知道他們想在實驗中研究什麼類型的動作。

We aim to improve the dataefficiency of learning expert-defined behaviors.
我們的目標是提高學習專家定義行為的數據效率。

Representation Learning. 表徵學習。

Visual representation learning has made great progress in effective representations images and videos [17, 16, 7, 29, 25, 19, 38].
視覺表示學習在有效表示圖像和視頻方面取得了很大進展 [17, 16, 7, 29, 25, 19, 38]。

Selfsupervised signals are often used to train this visual representation, such as learning relative positions of image patches [11], predicting image rotations [16], predicting future patches [29], and constrastive learning on augmented images [7].
自監督信號通常用於訓練這種視覺表示，例如學習圖像塊的相對位置 [11]、預測圖像旋轉 [16]、預測未來的塊 [29] 以及對增強圖像的解釋性學習 [7]。

Compared to visual data, trajectory data is significantly lower dimensional in each frame, and techniques from visual representation learning often cannot be applied directly.
與視覺數據相比，軌跡數據在每一幀中的維數要低得多，視覺表徵學習的技術往往不能直接應用。

For example, while we can create image patches that represent the same visual class, it is difficult to select a partial set of keypoints that represent the same behavior.
例如，雖然我們可以創建代表相同視覺類的圖像塊，但很難選擇代表相同行為的部分關鍵點集。

Our framework builds upon these approaches to learn effective representations for behavioral data.
我們的框架基於這些方法來學習行為數據的有效表示。

We investigate different decoder tasks in order to learn an effective behavior representation.
我們研究不同的解碼器任務以學習有效的行為表示。

One decoder task that we investigate is self-decoding: the reconstruction of input trajectories using generative modeling.
我們研究的一項解碼器任務是自解碼：使用生成建模重建輸入軌跡。

Generative modeling has previously been applied to learn representations for visual data [45, 38, 29] and language modeling [31]; for trajectory data, we use imitation learning [40, 44, 43] to train our trajectory representation.
生成式建模以前已被應用於學習視覺數據的表示 [45, 38, 29] 和語言建模 [31]； 對於軌跡數據，我們使用模仿學習 [40, 44, 43] 來訓練我們的軌跡表示。

The other tasks in our multitask self-supervised learning framework are created by domain experts using task

programming (Section 3.2).
我們的多任務自監督學習框架中的其他任務是由領域專家使用任務編程創建的（第 3.2 節）。

This idea of using a human-provided function as part of training has been studied for training set creation [33, 32], and controllable trajectory generation [43].
這種使用人類提供的函數作為訓練一部分的想法已經被研究用於訓練集創建 [33, 32] 和可控軌跡生成 [43]。

Our work explores these additional decoder tasks to further improve the learned representation over the generative loss alone.
我們的工作探索了這些額外的解碼器任務，以進一步改進學習表示而不是生成損失。

Multi-Task Self-Supervised Learning. 多任務自監督學習。

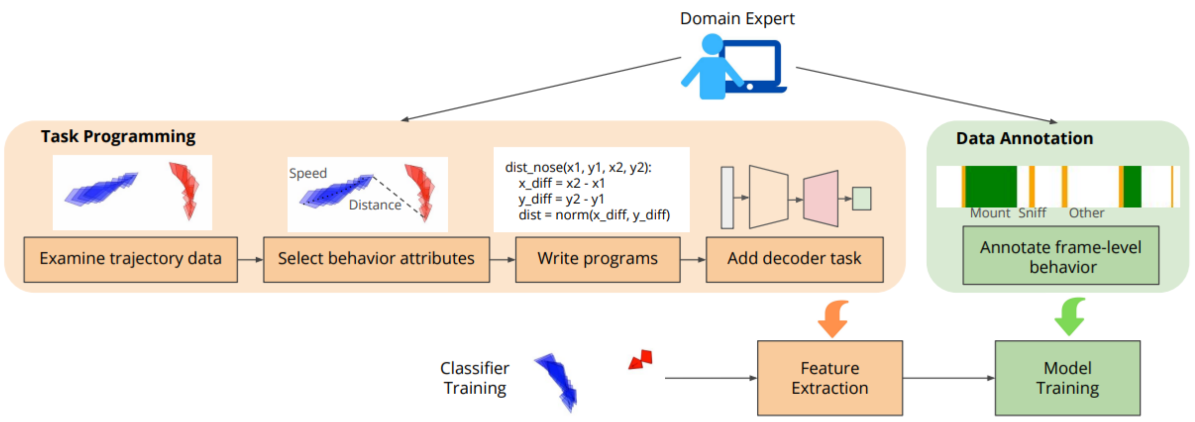
We jointly optimize a family of self-supervised tasks in an encoder-decoder setup, making this work an example of multitask self-supervised learning.
我們在編碼器解碼器設置中聯合優化了一系列自監督任務，使這項工作成為多任務自監督學習的一個例子。

Multi-task self-supervised learning has been applied to other domains such as visual data [12, 25], accelerometer recordings [35], audio [34] and multi-modal inputs [37, 30].
多任務自監督學習已應用於其他領域，例如視覺數據 [12, 25]、加速度計記錄 [35]、音頻 [34] 和多模態輸入 [37, 30]。

Generally in each of these domains, tasks are defined ahead of time, as is the case for tasks such as frame reconstruction, colorization, finding relative position of image patches, and video-audio alignment.
通常在這些領域中的每一個領域，任務都是提前定義的，例如幀重建、著色、查找圖像塊的相對位置和視頻-音頻對齊等任務。

Most of these tasks are designed for image or video data, and cannot be applied directly to trajectory data.
大多數這些任務是為圖像或視頻數據設計的，不能直接應用於軌跡數據。

To construct tasks for trajectory representation learning, we propose that domain experts can use task programming to engineer decoder tasks and encode structural knowledge.
為了構建用於軌跡表示學習的任務，我們建議領域專家可以使用任務編程來設計解碼器任務並編碼結構知識。



```
dist_nose(x1, y1, x2, y2):
  x_diff = x2 - x1
  y_diff = y2 - y1
  dist = norm(x_diff, y_diff)
```

Figure 2. **Task Programming and Data Annotation for Classifier Training.** Domain experts can choose between doing task programming and/or data annotation. Task programming is the process for domain experts to engineer decoder tasks for representation learning. The programs enable learning of annotation-sample efficient trajectory features to improve performance instead of additional annotations.

Figure 2. Task Programming and Data Annotation for Classifier Training.
圖 2. 分類器訓練的任務編程和數據註釋。

Domain experts can choose between doing task programming and/or data annotation.
領域專家可以選擇進行任務編程和/或數據註釋。

Task programming is the process for domain experts to engineer decoder tasks for representation learning.
任務編程是領域專家為表徵學習設計解碼器任務的過程。

The programs enable learning of annotation-sample efficient trajectory features to improve performance instead of additional annotations.
這些程序可以學習註釋樣本有效的軌跡特徵，以提高性能而不是額外的註釋。

3. Methods 方法

We introduce Trajectory Embedding for Behavior Analysis (TREBA), a method to learn an annotation-sample efficient trajectory representation using self-supervision and auxiliary decoder tasks engineered by domain experts.
我們介紹了用於行為分析的軌跡嵌入 (TREBA)，這是一種使用領域專家設計的自監督和輔助解碼器任務來學習註釋樣本有效軌跡表示的方法。

Figure 2 provides an overview of the expert’s role.
圖 2 概述了專家的角色。

In our framework, domain experts replace (a significant amount of) time-consuming manual annotation with the construction of a small number of programmed tasks, reducing total expert effort.
在我們的框架中，領域專家通過構建少量程序化任務來替代（大量）耗時的手動註釋，從而減少了專家的總工作量。

Each task places an additional constraint on the learned trajectory embedding.
每個任務都對學習到的軌跡嵌入施加了額外的約束。
TREBA uses the expert-programmed tasks based on a multi-task self-supervised learning approach, outlined in Figure 3.
TREBA 使用基於多任務自監督學習方法的專家編程任務，如圖 3 所示。

To learn task-relevant low-dimensional representations of pose trajectories, we train a network jointly on (1) reconstruction of the input trajectory (Section 3.1) and (2) expert-programmed decoder tasks (Section 3.3).
為了學習姿態軌蹟的任務相關低維表示，我們在（1）輸入軌蹟的重建（3.1 節）和（2）專家編程的解碼器任務（3.3 節）上聯合訓練網絡。

The learned representation can then be used as input to behavior modeling tasks, such as behavior classification.
然後可以將學習到的表示用作行為建模任務的輸入，例如行為分類。

3.1. Trajectory Representations 軌跡表示

Let D be a set of N unlabelled trajectories.
令 D 是一組 N 個未標記的軌跡。

Each trajectory is a sequence of states τ = {(st)}T t=1, where the state si at timestep i corresponds to the location or pose of the agents at that timestep.
每個軌跡是一系列狀態 τ = {(st)}T t=1，其中時間步 i 的狀態 si 對應於該時間步的代理的位置或姿態。

In this study, we divide trajectories from longer recordings into segments of length T, but in general trajectory length can vary.
在這項研究中，我們將來自較長記錄的軌跡劃分為長度為 T 的片段，但通常軌跡長度可能會有所不同。

For multiple agents, the keypoints of each agent is stacked at each timestep.
對於多個代理，每個代理的關鍵點在每個時間步堆疊。

Before we introduce our expert-programmed tasks,
在介紹我們的專家編程任務之前，

we will use trajectory reconstruction as an initial self-supervised task.
我們將使用軌跡重建作為初始自監督任務。

Given a history of agent states, we would like our model to predict the next state.

給定代理狀態的歷史，我們希望我們的模型預測下一個狀態。

This task is usually studied with sequential generative models.
此任務通常使用順序生成模型進行研究。

We used trajectory variational autoencoders (TVAEs) [9, 43] to embed the input trajectory using an RNN encoder, q φ, and an RNN decoder, p θ, to predict the next state.
我們使用軌跡變分自動編碼器 (TVAE) [9, 43] 使用 RNN 編碼器 q φ 和 RNN 解碼器 p θ 嵌入輸入軌跡，以預測下一個狀態。

The TVAE loss is:
TVAE 損失為：

$$\mathcal{L}^{\text{tvae}} = \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T} -\log(p_\theta(s_{t+1}|s_t,\mathbf{z}))\right] + D_{KL}(q_\phi(\mathbf{z}|\tau)||p_\theta(\mathbf{z})). \quad (1)$$

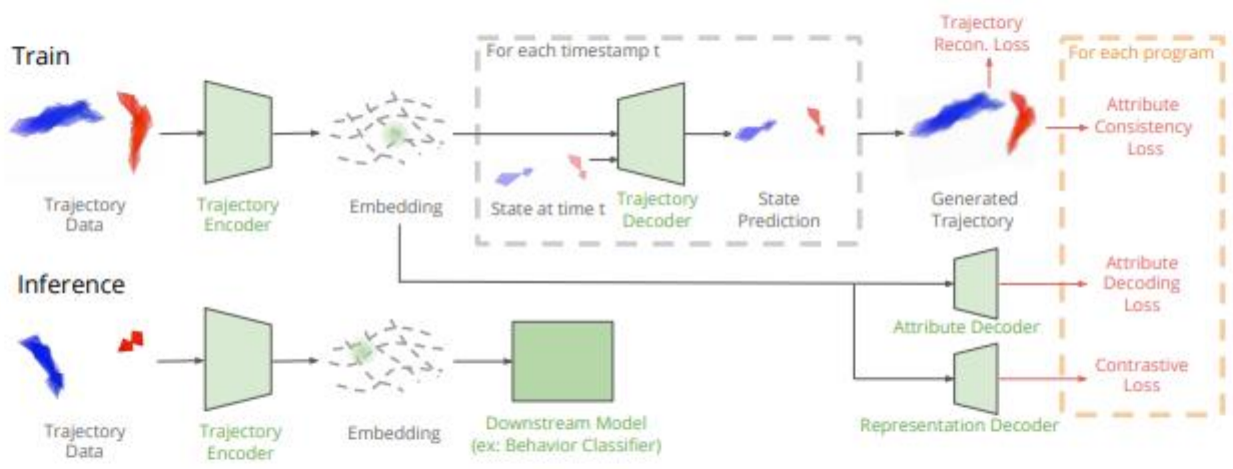


Figure 3. **TREBA Training and Inference Pipelines.** During training, we use trajectory self-decoding and the programmed decoder tasks to train the trajectory encoder. The learned representation is used for downstream tasks such as behavior classification.

Figure 3. TREBA Training and Inference Pipelines.
圖 3. TREBA 訓練和推理管道。

During training, we use trajectory self-decoding and the programmed decoder tasks to train the trajectory encoder.
在訓練期間，我們使用軌跡自解碼和編程的解碼器任務來訓練軌跡編碼器。

The learned representation is used for downstream tasks such as behavior classification.
學習到的表示用於下游任務，例如行為分類。

We use a prior distribution p(z) on z to regularize the learned embeddings; in this study, our prior is the unit Gaussian.
我們在 z 上使用先驗分佈 p(z) 來正則化學習到的嵌入； 在這項研究中，我們的先驗是單位高斯。

By optimizing for the TVAE loss only, we learn an unsupervised version of TREBA.
通過僅優化 TVAE 損失，我們學習了無監督版本的 TREBA。

When performing subsequent behavior modeling tasks such as classification, we use the embedding mean, z.
在執行分類等後續行為建模任務時，我們使用嵌入均值 z。

3.2. Task Programming 任務編程

Task programming is the process by which domain experts create decoder tasks for trajectory self-supervised learning.
任務編程是領域專家為軌跡自監督學習創建解碼器任務的過程。

This process consists of selecting attributes from trajectory data, writing programs, and creating decoder tasks based on the programs (Figure 2).
該過程包括從軌跡數據中選擇屬性、編寫程序以及基於程序創建解碼器任務（圖 2）。

Here, domain experts are people with specialized knowledge for studying behavior, such as neuroscientists or sports analysts.
在這裡，領域專家是具有研究行為的專業知識的人，例如神經科學家或運動分析師。

To start, domain experts identify attributes from trajectory data relevant to the behaviors of interest under study.
首先，領域專家從與研究中感興趣的行為相關的軌跡數據中識別屬性。

Behavior attributes capture information that is likely relevant to agent behavior, but is not explicitly included in the trajectory states {(st)}Tt =1.
行為屬性捕獲可能與代理行為相關的信息，但未明確包含在軌跡狀態 {(st)}Tt =1 中。

These attributes represent structured knowledge that domain experts are implicitly or explicitly considering for behavior analysis, such as the distance between two agents, agent velocity, or the relative positioning of agent body parts.
這些屬性代表了領域專家在進行行為分析時隱式或顯式考慮的結構化知識，例如兩個代理之間的距離、代理速度或代理身體部位的相對位置。

Next, domain experts write a program to compute these attributes on trajectory data, which can be done with existing tools such as MARS [36] or SimBA [28].
接下來，領域專家編寫一個程序來計算軌跡數據的這些屬性，這可以使用現有的工具完成，例如

MARS [36] 或 SimBA [28]。

Algorithm 1 shows a sample program from the mouse social behavior domain, for measuring the “facing angle” between a pair of interacting mice.
算法 1 顯示了一個來自鼠標社交行為領域的示例程序，用於測量一對交互鼠標之間的“面對角” 。

Each program can be used to construct decoder tasks for self-supervised learning (Section 3.3).
每個程序都可用於構建用於自監督學習的解碼器任務（第 3.3 節）。

Our framework is inspired by the data programming paradigm [33], which applies programs to training set creation.
我們的框架受到數據編程範式 [33] 的啟發，該範式將程序應用於訓練集創建。

In comparison, our framework uses task programming to unify expert-engineered programs, which encode structured expert knowledge, with representation learning.
相比之下，我們的框架使用任務編程來統一專家設計的程序，這些程序對結構化的專家知識進行編碼，與表示學習。

Algorithm 1: Sample Program for Facing Angle

```
Input: centroid of mouse 1 $(x_1, y_1)$, centroid of
  mouse 2 $(x_2, y_2)$, heading of mouse 1 $(\phi_1)$
$x_{\text{diff}} = x_2 - x_1$
$y_{\text{diff}} = y_2 - y_1$
$\theta = \arctan(y_{\text{diff}}, x_{\text{diff}})$
Return $\theta - \phi_1$
```

Domain	Behavior Attributes
Mouse	Facing Angle Mouse 1 and 2, Speed Mouse 1 and 2 Nose-Nose Distance, Nose-Tail Distance, Head-Body Angle Mouse 1 and 2 Nose Movement Mouse 1 and 2
Fly	Speed Fly 1 and 2, Fly-Fly Distance Angular Speed Fly 1 and 2, Facing Angle Fly 1 and 2 Min and Max Wing Angles Fly 1 and 2 Major/Minor Axis Ratio Fly 1 and 2

Table 1. **Behavior Attributes used in Task Programming.** We base our programmed tasks in our experiments on these behavior attributes from domain experts in each domain.

Algorithm 1: Sample Program for Facing Angle
演算法 1：面向角的示例程序

Table 1. Behavior Attributes used in Task Programming.
表 1. 任務編程中使用的行為屬性。

We base our programmed tasks in our experiments on these behavior attributes from domain experts in each domain.
我們在實驗中的編程任務基於來自每個領域的領域專家的這些行為屬性。

Working with domain experts in behavioral neuroscience, we created a set of programs to use in studying our approach.
我們與行為神經科學領域的專家合作，創建了一組程序用於研究我們的方法。

The selected programs are a subset of behavior attributes in [36] (for mouse datasets) and a subset of behavior attributes in [15] (for fly datasets).
選定的程序是 [36] 中的行為屬性子集（對於鼠標數據集）和 [15] 中的行為屬性子集（對於蒼蠅數據集）。

We list the programs used in Table 1, and provide more details about the programs in the Supplementary Material.
我們列出了表 1 中使用的程序，並在補充材料中提供了有關程序的更多詳細信息。

3.3. Learning Algorithm 學習演算法

We develop a method to incorporate the programs from domain experts as additional learning signals for TREBA.
我們開發了一種方法，將領域專家的程序作為 TREBA 的額外學習信號。

We consider the following three approaches:
(1) enforcing attribute consistency in generated trajectories (Section 3.3.1),
(2) performing attribute decoding directly (Section 3.3.2),
(3) applying contrastive loss based on program supervision (Section 3.3.3).
我們考慮以下三種方法：
(1) 在生成的軌跡中強制執行屬性一致性（第 3.3.1 節），
(2) 直接執行屬性解碼（第 3.3.2 節），
(3) 應用基於程序監督的對比損失（第 3.3.3 節）。

Each of these methods applies a different loss on the low-dimensional representation z of trajectory τ.
這些方法中的每一種都在軌跡 τ 的低維表示 z 上應用不同的損失。

Any combinations of these decoding tasks can be combined with self-decoding from Section 3.1 to inform the trajectory embedding z.
這些解碼任務的任何組合都可以與第 3.1 節中的自解碼相結合，以告知軌跡嵌入 z。

3.3.1 Attribute Consistency 屬性一致性

Let λ be a set of M domain-expert-designed functions measuring agent behavior attributes, such as agent velocity or facing angle.
令 λ 是一組 M 個領域專家設計的函數，用於測量代理行為屬性，例如代理速度或面對角度。

Recall that each λj , j = 1 … M takes as input a trajectory τ , and returns some expert-designed attribute λj(τ) computed from that trajectory.
回想一下，每個 λj , j = 1 … M 將軌跡 τ 作為輸入，並返回一些從該軌跡計算出的專家設計的屬性 λj(τ)。

For λj designed for a single frame, we apply the function to the center frame of τ .
對於為單幀設計的 λj，我們將函數應用於 τ 的中心幀。

Attribute consistency aims to maintain the same behavior attribute labels for the generated trajectory as the original.
屬性一致性旨在為生成的軌跡保持與原始軌跡相同的行為屬性標籤。

Let τ ~ be the trajectory generated by the TVAE given the same initial condition as and encoding z.
給定與編碼 z 相同的初始條件，讓 τ ~ 是由 TVAE 生成的軌跡。

The attribute consistency loss is:
屬性一致性損失為：

$$\mathcal{L}^{\text{attr}} = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{j=1}^{M} \mathbb{1}(\lambda_j(\tilde{\tau}) \neq \lambda_j(\tau)) \right]. \qquad (2)$$

Here, we show the loss for categorical λj , but in general, λj can be continuous and any loss measuring differences between λj(τ ~) and λj(τ) applies, such as mean squared error.
在這裡，我們展示了分類 λj 的損失，但一般來說，λj 可以是連續的，並且任何測量 λj(τ ~) 和 λj(τ) 之間差異的損失都適用，例如均方誤差。

We do not require λ to always be differentiable, and we use the differentiable approximation introduced in [43] to handle non-differentiable λ .
我們不要求 λ 總是可微的，我們使用 [43] 中引入的可微近似來處理不可微的 λ 。

3.3.2 Attribute Decoding 屬性解碼

Another option is to decode each attribute λj(τ) directly from the learned representation z.

另一種選擇是直接從學習到的表示 z 解碼每個屬性 λj(τ)。

Here we apply a shallow decoder f to the learned representation, with decoding loss:
在這裡，我們將淺層解碼器 f 應用於學習到的表示，具有解碼損失：

$$\mathcal{L}^{\text{decode}} = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{j=1}^{M} \mathbb{1}(f(q_\phi(\mathbf{z}_\mu|\tau)) \neq \lambda_j(\tau))\right]. \quad (3)$$

Similar to Eq. (2), we show the loss for categorical λj , however any type of λ may be used.
與 Eq. (2) 類似，我們展示了分類 λj 的損失，但是可以使用任何類型的 λ 。

3.3.3 Contrastive Loss 對比損失

Lastly, the programmed tasks can be used to supervise contrastive learning of our representation.
最後，編程任務可用於監督我們表示的對比學習。

For a trajectory τi, and for each λj , positive examples are those trajectories with the same attribute class under λj .
對於軌跡 τi 和每個 λj，正例是那些在 λj 下具有相同屬性類的軌跡。

For λj with continuous outputs, we create a discretized λ^j in which we apply fixed thresholds to divide the output space into classes.
對於具有連續輸出的 λj，我們創建了一個離散化的 λ^j，其中我們應用固定閾值將輸出空間劃分為類。

For our work, we apply two thresholds for each program such that our classes are approximately equal in size.
對於我們的工作，我們為每個程序應用兩個閾值，以便我們的類大小大致相等。

We apply a shallow decoder g to the learned representation, and let g = g(qϕ(zμ|τ)) represent the decoded representation.
我們將淺層解碼器 g 應用於學習到的表示，並讓 g = g(qφ(zμ| τ)) 表示解碼後的表示。

We then apply the contrastive loss:
然後我們應用對比損失：

$$\mathcal{L}^{\text{cntr.}} = \sum_{i=1}^{B}\sum_{j=1}^{M}\left[\frac{-1}{N_{pos(i,j)}}\sum_{k=1}^{B}\mathbb{1}_{i\neq k}\cdot\mathbb{1}_{\lambda_j(\tau_i)=\lambda_j(\tau_k)} \cdot \log\frac{\exp(\mathbf{g}_i\cdot\mathbf{g}_k/t)}{\sum_{l=1}^{N}\mathbb{1}_{i\neq l}\cdot\exp(\mathbf{g}_i\cdot\mathbf{g}_l/t)}\right], \quad (4)$$

where B is the batch size, Npos(i, j) is the number of positive matches for τ i with λ j , and t > 0 is a scalar temperature parameter.
其中 B 是批量大小，Npos(i, j) 是 τ i 與 λ j 的正匹配數，t > 0 是標量溫度參數。

Our form of contrastive loss supervised by task programming is similar to the contrastive loss in [24] supervised by human annotations.
我們由任務編程監督的對比損失形式類似於[24]中由人工註釋監督的對比損失。

A benefit of task programming is that the supervision from programs can be quickly and scalably applied to unlabelled datasets, as compared to expert supervision which can be time-consuming.
任務編程的一個好處是，與可能耗時的專家監督相比，程序監督可以快速且可擴展地應用於未標記的數據集。

We note that the unsupervised version of this contrastive loss is studied in [7], based on previous works such as [29].
我們注意到在 [7] 中研究了這種對比損失的無監督版本，基於以前的工作，如 [29]。

3.3.4 Data Augmentation 數據增強

We can perform data augmentation on trajectory data based on our expert-provided programs.
我們可以根據專家提供的程序對軌跡數據進行數據增強。

Given the set of all possible augmentations, we define to be the subset of augmentations that are attribute-preserving: that is, for all λ j in the set of programs, λ j(τ) = λ j(Λ m(τ)) for some augmentation Λ m ∈ Λ.
給定所有可能的增強集，我們定義為保留屬性的增強子集：也就是說，對於程序集中的所有 λj，對於某些增強 Λm，λ j(τ) = λj(Λm(τ)) ∈ Λ。

An example of a valid augmentation in the mouse domain is reflection of the trajectory data.
鼠標域中有效增強的一個示例是軌跡數據的反射。

All losses presented above can be extended with data augmentation, by replacing τ with Λ m (τ) in losses.
通過在損失中用 Λ m (τ) 替換 τ ，可以通過數據增強來擴展上述所有損失。

For contrastive loss, adding data augmentation corresponds to extending the batch size to 2B, with B samples

from the original and augmented trajectories.
對於對比損失，添加數據增強對應於將批量大小擴展到 2B，其中 B 個樣本來自原始軌跡和增強軌跡。

The augmentations we use in our experiments are reflections, rotations, translations, and a small Gaussian noise on the keypoints (mouse data only).
我們在實驗中使用的增強是反射、旋轉、平移和關鍵點上的小高斯噪聲（僅鼠標數據）。

In practice, we add the loss for each decoder with and without data augmentation.
在實踐中，我們為每個解碼器添加了帶有和不帶有數據增強的損失。

4. Experiments 實驗

4.1. Datasets 數據集

We work with datasets from behavioral neuroscience, where there are large-scale, expert-annotated datasets from scientific experiments.
我們使用來自行為神經科學的數據集，其中有來自科學實驗的大規模、專家註釋的數據集。

We study behavior for the laboratory mouse and the fruit fly, two of the most common model organisms in behavioral neuroscience.
我們研究了實驗室小鼠和果蠅的行為，這兩種行為神經科學中最常見的模式生物。

For each organism, we first train TREBA using large unannotated datasets: for the mouse domain we use an in-house dataset comprised of approximately 100 hours of recorded diadic social interactions (Mouse100), while for the fly domain we use the Fly vs. Fly dataset [15] without annotations.
對於每個生物，我們首先使用大型未註釋數據集訓練 TREBA：對於鼠標域，我們使用由大約 100 小時記錄的二元社交互動 (Mouse100) 組成的內部數據集，而對於蒼蠅域，我們使用 Fly vs. Fly 沒有註釋的數據集 [15]。

After pre-training TREBA, we evaluate the suitability of our trajectory representation for supervised behavior classification (classifying frame-level behaviors on continuous trajectory data), on three additional datasets:
在對 TREBA 進行預訓練後，我們在三個額外的數據集上評估了我們的軌跡表示對監督行為分類（在連續軌跡數據上對幀級行為進行分類）的適用性：

MARS.

The MARS dataset [36] is a recently released mouse social behavior dataset collected in the same conditions as Mouse100.
MARS 數據集 [36] 是最近發布的鼠標社交行為數據集，在與 Mouse100 相同的條件下收集。

The dataset is annotated by neurobiologists on a frame-by-frame basis for three behaviors: sniff, attack, and mount.
該數據集由神經生物學家逐幀註釋，用於三種行為：嗅探、攻擊和安裝。

We use the provided train, validation, and test split (781k, 352k, and 184k frames respectively).
我們使用提供的訓練、驗證和測試分割（分別為 781k、352k 和 184k 幀）。

Trajectories are extracted by the MARS tracker [36].
軌跡由 MARS 跟踪器 [36] 提取。

CRIM13.
CRIM13 [4] is a second mouse social behavior dataset manually annotated on a frame-by-frame basis by experts.
CRIM13.
CRIM13 [4] 是由專家逐幀手動註釋的第二個鼠標社交行為數據集。

To extract trajectories, we use a version of the the MARS tracker [36] fine-tuned on pose annotations on CRIM13.
為了提取軌跡，我們使用了一個版本的 MARS 跟踪器 [36]，在 CRIM13 上的姿勢註釋上進行了微調。

We select a subset of videos from which trajectories can be reliably detected for a train, validation and test split of 407k, 96k, and 142k frames respectively.
我們選擇了一個視頻子集，從中可以可靠地檢測到 407k、96k 和 142k 幀的訓練、驗證和測試分割的軌跡。

We evaluated classifier performance on the same three behaviors studied in MARS (sniff, attack, mount).
我們評估了在 MARS 中研究的相同三種行為（嗅探、攻擊、裝載）的分類器性能。

CRIM13 is a useful test of the robustness of TREBA trained on Mouse100, as the recording conditions in CRIM13 (image resolution 640   480, frame rate 25Hz, and non-centered cage location) are different from those of Mouse100 (image resolution 1024 570, frame rate 30Hz, and centered cage location).
CRIM13 是在 Mouse100 上訓練的 TREBA 魯棒性的有用測試，因為 CRIM13 中的記錄條件（圖像分辨率 640 480，幀率 25Hz，非中心籠位置）不同於 Mouse100（圖像分辨率 1024 570，幀 率 30Hz，籠子位置居中）。

Fly vs. Fly (Fly).

We use the Aggression and Courtship videos from the Fly dataset [15].
我們使用 Fly 數據集 [15] 中的 Aggression 和 Courtship 視頻。

These videos record interactions between a pair of flies annotated on a frame-byframe basis for social

behaviors by domain experts.
這些視頻記錄了一對蒼蠅之間的互動，由領域專家逐幀註釋社會行為。

Our train, validation and test split has 1067k, 162k, 322k frames respectively.
我們的訓練、驗證和測試分割分別有 1067k、162k、322k 幀。

We use the trajectories tracked by [15] and evaluate on all behaviors with more than 1000 frames of annotations in the full training set (lunge, wing threat, tussle, wing extension, circle, copulation).
我們使用 [15] 跟踪的軌跡，並在完整的訓練集（弓步、機翼威脅、爭鬥、機翼伸展、圓圈、交配）中評估具有超過 1000 幀註釋的所有行為。

4.2. Training and Evaluation Procedure 培訓和評估程序

We use the attribute consistency loss (Section 3.3.1) and contrastive loss (Section 3.3.3) to train TREBA using programs.
我們使用屬性一致性損失（第 3.3.1 節）和對比損失（第 3.3.3 節）來使用程序訓練 TREBA。

With the same programs, we find that different loss combinations result in similar performance, and that the combination of consistency and contrastive losses performs the best overall.
使用相同的程序，我們發現不同的損失組合會導致相似的性能，並且一致性和對比損失的組合總體上表現最好。

The results for all loss combinations are provided in the Supplementary Material.
補充材料中提供了所有損耗組合的結果。

For the datasets in the mouse domain (MARS and CRIM13) we train TREBA on Mouse100, with 10 programs provided by mouse behavior domain experts.
對於鼠標域（MARS 和 CRIM13）中的數據集，我們在 Mouse100 上訓練 TREBA，並使用鼠標行為域專家提供的 10 個程序。

For the Fly dataset, we train TREBA on the training split of Fly without annotations, with 13 programs provided by fly behavior domain experts.
對於 Fly 數據集，我們使用 Fly 行為領域專家提供的 13 個程序在沒有註釋的 Fly 的訓練分割上訓練 TREBA。

The full list is in Table 1.
完整列表見表 1。

We then use the trained encoder, with pre-trained frozen weights, as a trajectory feature extractor over T = 21 frames, where the representation for each frame is computed using ten frames before and after the current frame.

然後，我們使用經過訓練的編碼器和預先訓練的凍結權重，作為 T = 21 幀的軌跡特徵提取器，其中每幀的表示是使用當前幀前後的 10 幀計算的。

We evaluate our classifiers, with and without TREBA features, using Mean Average Precision (MAP).
我們使用平均平均精度 (MAP) 評估我們的分類器，有和沒有 TREBA 特徵。

We compute the mean over behaviors of interest with equal weighting.
我們以相同的權重計算感興趣的行為的平均值。

Our classifiers are shallow fully-connected neural networks on the input features.
我們的分類器是輸入特徵上的淺層全連接神經網絡。

To determine the relationship between classifier performance and training set size, we sub-sample the training data by randomly sampling trajectories (with lengths of 100 frames) to achieve a desired fraction of the training set size.
為了確定分類器性能和訓練集大小之間的關係，我們通過隨機採樣軌跡（長度為 100 幀）對訓練數據進行子採樣，以達到訓練集大小的所需比例。

Sampling was performed to achieve a similar class distribution as the full training set.
執行採樣以實現與完整訓練集類似的類分佈。

We train each classifier nine times over three different random selections of the training data for each training fraction(1%, 2%, 5%, 10%, 25%, 50%, 75%, 100%).
我們針對每個訓練分數（1%、2%、5%、10%、25%、50%、75%、100%）對訓練數據的三個不同隨機選擇對每個分類器訓練九次。

Additional implementation details are in the Supplementary Material.
其他實施細節在補充材料中。

4.3. Main Results 主要結果

We evaluate the data efficiency of our representation for supervised behavior classification, by training a classifier to predict behavior labels given both our learned representation and one of either (1) raw keypoints or (2) domain-specific features designed by experts.
我們通過訓練分類器來預測給定我們學習的表示和 (1) 原始關鍵點或 (2) 專家設計的特定領域特徵之一的行為標籤，來評估我們的表示對監督行為分類的數據效率。

The TREBA+keypoints evaluation allows us to test the effectiveness of our representation without other hand-designed features, while the TREBA+features evaluation is closer to most potential use cases.
TREBA+關鍵點評估允許我們在沒有其他手工設計特徵的情況下測試我們表示的有效性，而 TREBA+特徵評估更接近大多數潛在用例。

The domain-specific features for mice are the trajectory features from [36] and features for flies are the trajectory features from [4].
小鼠的領域特定特徵是來自 [36] 的軌跡特徵，蒼蠅的特徵是來自 [4] 的軌跡特徵。

The input features are a superset of the programs we use in Table 1.
輸入特徵是我們在表 1 中使用的程序的超集。

Our representation is able to improve the data efficiency for both keypoints and domain-specific features, over all evaluated amounts of training data availability (Figure 4).
我們的表示能夠提高關鍵點和特定領域特徵的數據效率，超過所有評估的訓練數據可用性量（圖 4）。

We discuss each dataset below:
我們在下面討論每個數據集：

MARS.

Our representation significantly improves classification performance over keypoints alone (Figure 4 A1).
我們的表示比單獨的關鍵點顯著提高了分類性能（圖 4 A1）。

We achieve the same performance as the full baseline training using only between 1% and 2% of the data.
我們僅使用 1% 到 2% 的數據就獲得了與完整基線訓練相同的性能。

While this result is partially because our representation contains temporal information, we can also observe a significant increase in data efficiency in A2 compared to domain-specific features, which also contains temporal features.
我們僅使用 1% 到 2% 的數據就獲得了與完整基線訓練相同的性能。

Classifiers using TREBA has the same performance as the full baseline training set with around 5% 10% of data (i.e., 10 20improved annotation efficiency).
使用 TREBA 的分類器具有與完整基線訓練集相同的性能，大約有 5% 10% 的數據（即 10 20 提高了註釋效率）。

CRIM13.

We test the transfer learning ability of our representation on CRIM13, a dataset with different image properties than Mouse100, the training set of TREBA.
我們在 CRIM13 上測試了我們的表示的遷移學習能力，CRIM13 是一個具有與 TREBA 訓練集 Mouse100 不同圖像屬性的數據集。

Our representation achieves the same performance as the baseline training with keypoints using around 5% to

10% of the training data (Figure 4 B1).
我們的表示使用大約 5% 到 10% 的訓練數據（圖 4 B1）使用關鍵點實現了與基線訓練相同的性能。

With domain-specific features, TREBA uses 50% of the data annotation to have the same performance as the full training baseline (i.e., 2 improved annotation efficiency).
借助特定領域的特徵，TREBA 使用 50% 的數據註釋具有與完整訓練基線相同的性能（即提高了 2 個註釋效率）。

Our representation is able to generalize to a different dataset of the same organism.
我們的表示能夠推廣到同一生物體的不同數據集。

Fly.

When using keypoints only, our representation re-quires 10% of the data (Figure 4 C1) and for features, our representation requires 50% of the data (Figure 4 C2) to achieve the same performance as full baseline training.
僅使用關鍵點時，我們的表示需要 10% 的數據（圖 4 C1），而對於特徵，我們的表示需要 50% 的數據（圖 4 C2）才能實現與完整基線訓練相同的性能。

This corresponds to 2improved annotation efficiency.
這對應於 2 改進的註釋效率。

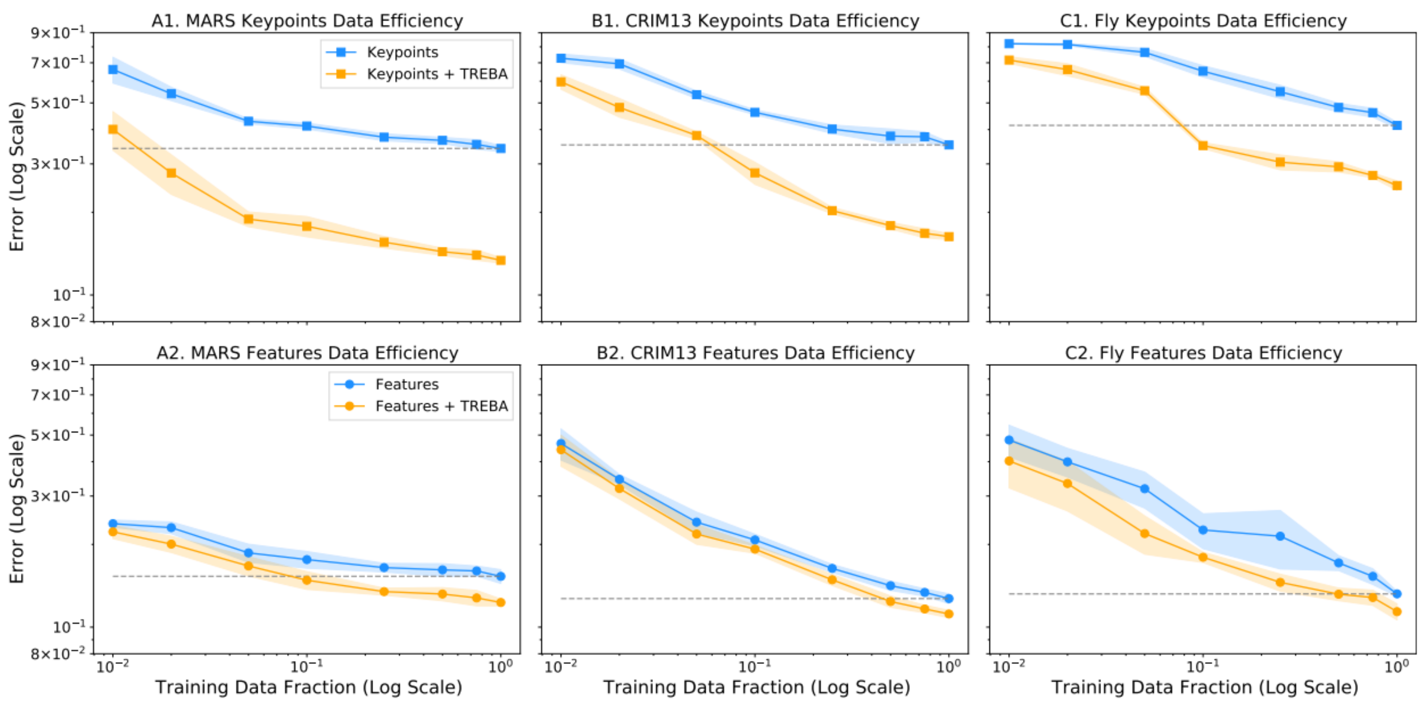


Figure 4. **Data Efficiency for Supervised Classification.** Training data fraction vs. classifier error on MARS (left), CRIM13 (middle) and fly (right). The blue lines represent performance with baseline keypoints and features, and the orange lines are with TREBA. The shaded regions correspond to the classifier standard deviation over nine repeats. The gray dotted line marks the best observed classifier performance when trained on the baseline features (using the full training set). Note the log scale on both the x and y axes.

Figure 4. Data Efficiency for Supervised Classification.

圖 4. 監督分類的數據效率。

Training data fraction vs. classifier error on MARS (left), CRIM13 (middle) and fly (right).
MARS（左）、CRIM13（中）和 fly（右）上的訓練數據分數與分類器誤差。

The blue lines represent performance with baseline keypoints and features, and the orange lines are with TREBA.
藍線代表基線關鍵點和特徵的性能，橙色線代表 TREBA。

The shaded regions correspond to the classifier standard deviation over nine repeats.
陰影區域對應於九次重複的分類器標準偏差。

The gray dotted line marks the best observed classifier performance when trained on the baseline features (using the full training set).
灰色虛線表示在基線特徵（使用完整訓練集）上訓練時觀察到的最佳分類器性能。

Note the log scale on both the x and y axes.
注意 x 軸和 y 軸上的對數刻度。

4.4. Model Ablations 模型消融

We perform the following model ablations to better characterize our approach.
我們執行以下模型消融以更好地表徵我們的方法。

In this section, percentage error reduction relative to baseline is averaged over all training fractions.
在本節中，相對於基線的錯誤減少百分比是所有訓練分數的平均值。

Additional results are in the Supplementary Material.
其他結果在補充材料中。

Varying Programmed Tasks.
不同的編程任務。

We test the performance of TREBA trained with each single program provided by the domain experts in Table 1, and the average, best, and worst performance is visualized in Figure 5.
我們測試了使用表 1 中領域專家提供的每個程序訓練的 TREBA 的性能，平均、最佳和最差性能如圖 5 所示。

On average, representations learned from a single program is better than using features alone, but using all provided programs further improves performance.
平均而言，從單個程序中學習的表徵比單獨使用特徵要好，但使用所有提供的程序可以進一步提高性

能。

For a single program, there could be a large variation in performance depending on the selected program (Figure 5).
對於單個程序，根據所選程序的不同，性能可能會有很大差異（圖 5）。

While the best performing single program is close in classifier MAP to using all programs, the worst performing program may increase error, as in MARS and CRIM13.
雖然在分類器 MAP 中性能最好的單個程序接近使用所有程序，但性能最差的程序可能會增加錯誤，如 MARS 和 CRIM13。

We further tested the performance using more programs.
我們使用更多程序進一步測試了性能。

In the mouse domain, we found that with three randomly selected programs, the variation between runs is much smaller compared to single programs (Supplementary Material).
在鼠標域中，我們發現使用三個隨機選擇的程序，與單個程序（補充材料）相比，運行之間的差異要小得多。

With three programs, we achieve comparable average error reduction from baseline features to using all programs(MARS: 14.6% error reduction for 3 programs vs. 15.3% for all, CRIM13: 9.2% for 3 programs vs. 9.5% for all).
使用三個程序，我們實現了從基線特徵到使用所有程序的可比平均錯誤減少（MARS：3 個程序的錯誤減少 14.6% 對所有程序的 15.3%，CRIM13：3 個程序的 9.2% 對所有程序的 9.5%）。

For the fly domain, we found that we needed seven programs to achieve comparable performance (20.7% for 7 programs vs. 21.2% for all).
對於飛行域，我們發現我們需要 7 個程序才能達到可比的性能（7 個程序為 20.7%，所有程序為 21.2%）。

Varying Decoder Losses. 變化的解碼器損耗。

When the programmed tasks are fixed, decoder losses with different combinations of consistency (Section 3.3.1), decoding (Section 3.3.2), and contrastive (Section 3.3.3) loss are similar in performance (Supplementary Material).
當編程任務固定時，具有一致性（第 3.3.1 節）、解碼（第 3.3.2 節）和對比（第 3.3.3 節）損失的不同組合的解碼器損失在性能上相似（補充材料）。

Additionally, we evaluate the TREBA framework without programmed tasks, with decoder tasks using trajectory generation and unsupervised contrastive loss.
此外，我們在沒有編程任務的情況下評估 TREBA 框架，解碼器任務使用軌跡生成和無監督對比損失。

While self-supervised representations are also effective at reducing baseline error, we achieve the best classifier performance using TREBA with programmed tasks (Table 2).
雖然自監督表示在減少基線錯誤方面也很有效，但我們使用 TREBA 和編程任務實現了最佳分類器性能（表 2）。

Furthermore, we found that training trajectory representations without self-decoding, using the contrastive loss from [7, 8], resulted in less effective representations for classification (Supplementary Material).
此外，我們發現在沒有自解碼的情況下訓練軌跡表示，使用來自 [7, 8] 的對比損失，導致分類表示不太有效（補充材料）。

Data Augmentation. 數據增強。

We removed the losses using the data augmentation described in Section 3.3.4, and found that performance was slightly lower for all datasets than with augmentation.
我們使用第 3.3.4 節中描述的數據增強去除了損失，發現所有數據集的性能都比增強略低。

In particular, adding data augmentation decreases error by 1:2% on MARS, 2:5% on CRIM13, and 5:3% on Fly compared to without data augmentation.
特別是，與沒有數據增強的情況相比，添加數據增強在 MARS 上減少了 1:2%，在 CRIM13 上減少了 2:5%，在 Fly 上減少了 5:3%。

Pre-Training Variations 預訓練變化

The results shown for MARS was obtained with pre-training TREBA on Mouse100, a large in-house mouse dataset with the same image prop-erties as MARS.
MARS 顯示的結果是通過在 Mouse100 上預訓練 TREBA 獲得的，Mouse100 是一個大型內部鼠標數據集，具有與 MARS 相同的圖像屬性。

Figure 6 demonstrates the effect of varying TREBA training data amount with TVAE only and with programs.
圖 6 展示了僅使用 TVAE 和使用程序改變 TREBA 訓練數據量的效果。

For both keypoints and features, we observe that TVAE (MARS) has the largest error.
對於關鍵點和特徵，我們觀察到 TVAE (MARS) 的誤差最大。

We see that error can be decreased by either adding more data (features + TVAE (Mouse100) with 3:9% decrease) or adding task programming (features + Programs (MARS) with 4:4% decrease).
我們看到可以通過添加更多數據（特徵 + TVAE (Mouse100) 減少 3:9%）或添加任務編程（特徵 + 程序（MARS）減少 4:4%）來減少錯誤。

Adding both more data and task programming results in an average decrease of 5:7% error relative to TVAE

(MARS) and the lowest average error.
添加更多數據和任務編程導致相對於 TVAE (MARS) 平均減少 5:7% 的錯誤和最低的平均錯誤

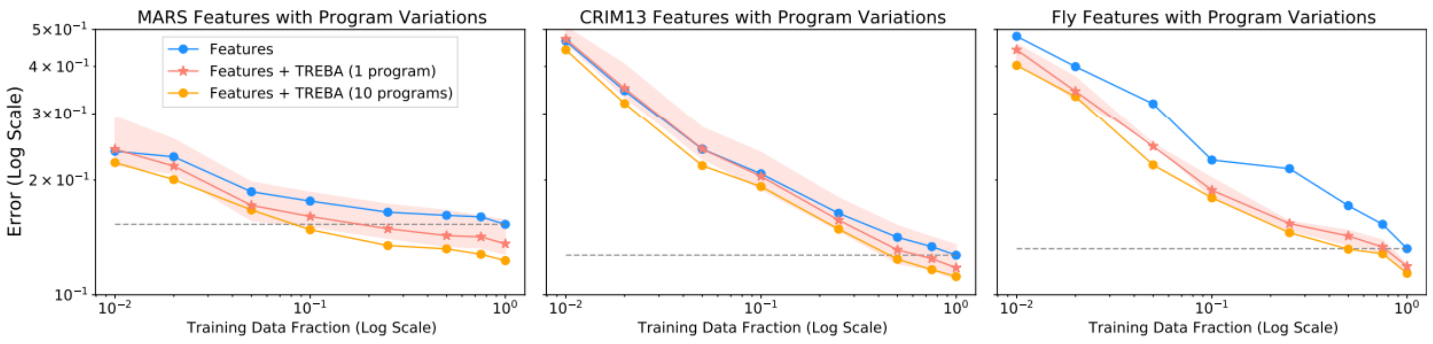


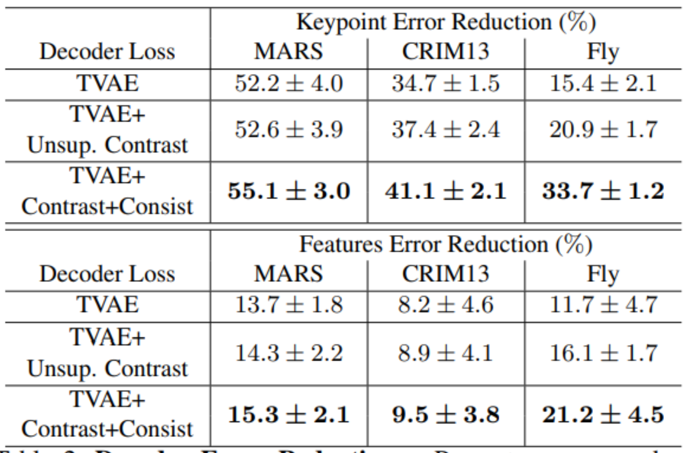
Figure 5. **Varying Programmed Tasks.** Effect of varying number of programmed tasks on classifier data efficiency. The shaded region corresponds to the best and worst classifiers trained using a single programmed task from Table 1. The grey dotted line corresponds to the value where the baseline features achieve the best performance (using the full training set).

Figure 5. Varying Programmed Tasks.
圖 5. 不同的編程任務。

Effect of varying number of programmed tasks on classifier data efficiency.
不同數量的編程任務對分類器數據效率的影響。

The shaded region corresponds to the best and worst classifiers trained using a single programmed task from Table 1.
陰影區域對應於使用表 1 中的單個編程任務訓練的最佳和最差分類器。

The grey dotted line corresponds to the value where the baseline features achieve the best performance (using the full training set).
灰色虛線對應基線特徵達到最佳性能的值（使用完整訓練集）。



Decoder Loss	Keypoint Error Reduction (%) MARS	CRIM13	Fly
TVAE	52.2 ± 4.0	34.7 ± 1.5	15.4 ± 2.1
TVAE+ Unsup. Contrast	52.6 ± 3.9	37.4 ± 2.4	20.9 ± 1.7
TVAE+ Contrast+Consist	$\mathbf{55.1 \pm 3.0}$	$\mathbf{41.1 \pm 2.1}$	$\mathbf{33.7 \pm 1.2}$

Decoder Loss	Features Error Reduction (%) MARS	CRIM13	Fly
TVAE	13.7 ± 1.8	8.2 ± 4.6	11.7 ± 4.7
TVAE+ Unsup. Contrast	14.3 ± 2.2	8.9 ± 4.1	16.1 ± 1.7
TVAE+ Contrast+Consist	$\mathbf{15.3 \pm 2.1}$	$\mathbf{9.5 \pm 3.8}$	$\mathbf{21.2 \pm 4.5}$

Table 2. **Decoder Error Reductions.** Percentage error reduction relative to baseline keypoints and domain-specific features for training with different decoder losses for TREBA. The average is taken over all evaluated training fractions.

Table 2. Decoder Error Reductions.

表 2. 解碼器錯誤減少。

Percentage error reduction relative to baseline keypoints and domain-specific features for training with different decoder losses for TREBA.
相對於基線關鍵點和特定領域特徵的誤差減少百分比，用於使用 TREBA 的不同解碼器損失進行訓練。

The average is taken over all evaluated training fractions.
對所有評估的訓練分數取平均值。

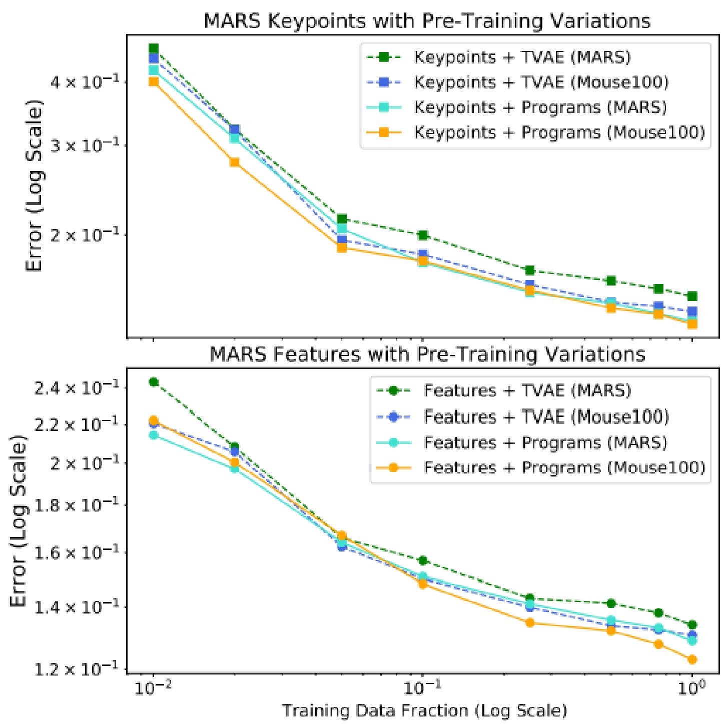


Figure 6. **Pre-Training Data Variations.** Effect of varying pre-training data on classifier data efficiency for the MARS dataset. “TVAE” corresponds to training TREBA with TVAE losses only, and “Programs” corresponds to training with all programs.

Figure 6. Pre-Training Data Variations. Effect of varying pretraining data on classifier data efficiency for the MARS dataset.
圖 6. 預訓練數據變化。 不同預訓練數據對 MARS 數據集分類器數據效率的影響。

“TVAE” corresponds to training TREBA with TVAE losses only, and “Programs” corresponds to training with all programs.
“TVAE” 對應於僅使用 TVAE 損失訓練 TREBA， “Programs” 對應於所有程序的訓練。

5. Conclusion 結論

We introduce a method to learn an annotation-sample efficient Trajectory Embedding for Behavior Analysis (TREBA).
我們介紹了一種學習註釋樣本高效軌跡嵌入行為分析（TREBA）的方法。

To train this representation, we study selfsupervised decoder tasks as well as decoder tasks with programmatic supervision, the latter created using task programming.
為了訓練這種表示，我們研究了自監督解碼器任務以及具有程序監督的解碼器任務，後者使用任務編程創建。

Our results show that TREBA can reduce annotation requirements by a factor of 10 for mice and 2 for flies.
我們的結果表明，TREBA 可以將小鼠的註釋需求減少 10 倍，蒼蠅的註釋需求減少 2 倍。

Our experiments on three datasets (two in mice and one in fruit flies) suggest that our approach is effective across different domains.
我們在三個數據集（兩個在小鼠中，一個在果蠅中）上的實驗表明，我們的方法在不同領域都是有效的。

TREBA is not restricted to animal behavior and may be applied to other domains where tracking data is expensive to annotate, such as in sports analytics.
TREBA 不僅限於動物行為，還可應用於其他領域，其中跟踪數據的註釋成本很高，例如體育分析。

Our experiments highlight, and quantify, the tradeoff between task programming and data annotation.
我們的實驗強調並量化了任務編程和數據註釋之間的權衡。

The choice of which is more effective will depend on the cost of annotation and the level of expert understanding in identifying behavior attributes.
選擇哪個更有效將取決於註釋的成本和識別行為屬性的專家理解水平。

Directions in creating tools to facilitate program creation and data annotation will help further accelerate behavioral studies.
創建工具以促進程序創建和數據註釋的方向將有助於進一步加速行為研究。

6. Acknowledgements 致謝

We would like to thank Tomomi Karigo at Caltech for providing the mouse dataset.
我們要感謝加州理工學院的 Tomomi Karigo 提供鼠標數據集。

The Simons Foundation (Global Brain grant 543025 to PP) generously supported this work, and this work is partially supported by NIH Award #K99MH117264 (to AK), NSF Award #1918839 (to YY), and NSERC Award #PGSD3-532647-2019 (to JJS).
西蒙斯基金會（全球大腦資助 543025 給 PP）慷慨地支持這項工作，這項工作得到了 NIH 獎 #K99MH117264（給 AK）、NSF 獎 #1918839（給 YY）和 NSERC 獎 #PGSD3-532647-2019 的部分支持（對 JJS）。