任务编程: 学习数据有效的行为表示

Jennifer J. Sun[1] Ann Kennedy[2] Eric Zhan[1] David J. Anderson[1] Yisong Yue[1] Pietro Perona[1]
[1]Caltech [2]Northwestern University

代码 & 项目地址: https://sites.google.com/view/task-programming

摘要

专业领域知识通常是准确标注训练集以进行深入分析所必需的，但从领域专家那里获取可能既繁琐又耗时。这个问题在自动行为分析中尤为突出，因为其中智能体的移动或我们感兴趣的动作是从视频跟踪数据中检测到的。为了减少标注工作，我们提出了一种基于多任务自监督学习的标注样本学习方法 TREBA，该方法可以有效地嵌入用于行为分析的轨迹。我们方法中的任务可以由领域专家通过我们称为“任务编程”的过程有效地设计，该过程使用程序显式地编码来自领域专家的结构化知识。通过将数据标注的时间用来构建少量编程任务，可以减少领域专家的总工作量。我们使用来自行为神经科学的数据来评估这种权衡，在行为神经科学中，专门的领域知识被用来识别行为。我们在小鼠和果蝇两个领域的三个数据集中展示了实验结果。使用来自 TREBA 的嵌入，与 SOTA 特征相比，我们在不影响准确性的情况下将标注工作减少了 10 倍。因此，我们的结果表明，任务编程和自监督可以成为减少领域专家工作标注工作的有效方法。

1. 简介

一个或多个智能体的行为分析是多个研究领域的核心要素，包括生物学 [36, 26]，自动驾驶 [6, 39]，体育分析 [42, 43]，和视频游戏 [20, 3]。在典型的工作流中，首先从视频的每一帧中提取智能体的位置和姿势，然后根据智能体的姿势和运动逐帧应用实验者定义的感兴趣行为的标签。自动量化与手动标注相比，除了减少人力外，可以产生更客观、精确和可伸缩的度量 [1, 10]。然而，训练行为检测模型可能是数据密集型的，手动标注行为通常需要专门的领域知识和高频时间标签，生成训练数据集的过程对于专家来说既耗时又费力。因此，需要减少领域

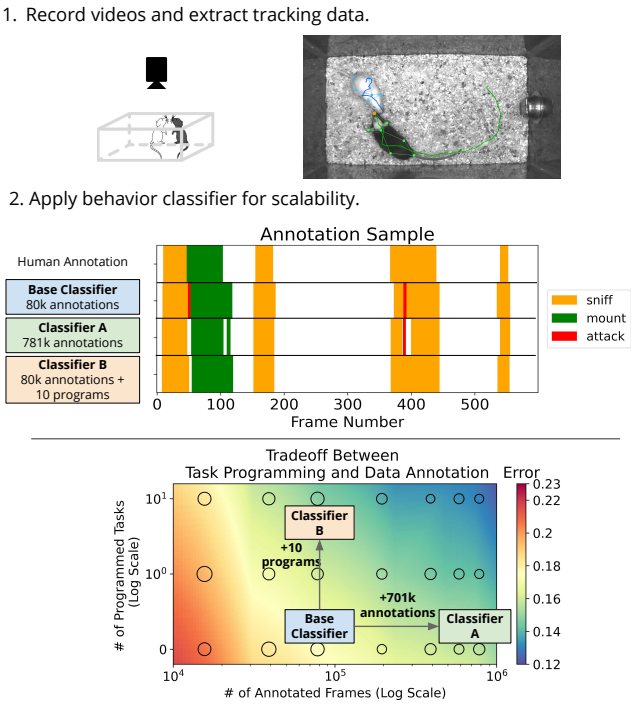

图 1. **方法概述**。Part 1: 典型的行为研究从提取视频中的跟踪数据开始。我们展示了每只小鼠的 7 个关键点和鼻关键点的轨迹。Part 2: 领域专家可以进行数据标注（分类器 A）或任务编程（分类器 B）以减少分类器误差。中间面板显示 30Hz 的带标注帧。底部图中的颜色表示圆形标记处分类器的误差表现 (完整结果见 Section 4.3)。圆形标记的大小表示误差的方差。

Correspondence to jjsun@caltech.edu.

专家标注工作的方法来加速行为研究。

我们研究提高分类器准确性的其他方法，而不仅仅是增加标注数据的绝对数量。我们提出了一个框架，它统一了：（1）自监督表示学习，以及（2）使用专家定义的程序对轨迹数据的显式结构化知识进行编码。领域专家可以有效地构建这些程序，因为每帧中的关键点轨迹通常是低维的，并且专家已经可以为轨迹数据手动设计有效的特征 [36, 28]。为了最好地利用这种结构化的专家知识，我们开发了一个框架来学习基于多任务自监督学习的轨迹表示，但尚未对轨迹数据进行充分探索。

我们的方法。我们的框架 Trajectory Embedding for Behavior Analysis (TREBA)，通过轨迹生成和一组解码器学习轨迹表示基于专家设计的程序的任务。受数据编程范式的启发 [33]，这些程序是由领域专家通过我们称为任务编程的过程创建的。任务编程是领域专家识别与研究中感兴趣的行为相关的轨迹属性、编写程序并将这些程序应用于表示学习的过程（Section 3.2）。解码器任务的这种灵活性使我们的框架适用于跨不同研究领域研究的各种行为和智能体。

专家工作的权衡。由于任务编程通常需要领域专家的时间，我们研究了进行任务编程和数据标注之间的权衡。我们比较了不同数量的带标注的训练数据和编程任务的行为分类性能。例如，对于图 1中所示的领域，领域专家可以通过标注 701k 个额外帧，相对于基本分类器减少 13% 的错误，或者通过在我们的框架中使用 10 个编程任务学习表示，他们可以减少 16% 的错误。我们的方法允许专家用少量编程任务代替大量标注。

我们在行为神经科学的两个领域研究我们的方法，即小鼠和果蝇的行为。我们这样选择是因为它需要专门的领域知识来进行数据标注，而数据效率对于领域专家来说很重要。此外，我们框架中的解码器任务可以由专家根据描述轨迹属性的简单函数有效地编程，以识别感兴趣的行为。例如，对于攻击 [36]等小鼠社交行为，重要的行为属性包括每只小鼠的速度和小鼠之间的距离。相应的任务可能是从学习到的表示中解码这些属性。

我们的贡献是：

- 我们引入任务编程作为领域专家减少标注工作和编码结构知识的有效方法。我们开发了一种新的方法来学习标注样本，并使用自我监督和程序监督来实现有效的轨迹表示。
- 我们研究了任务编程、数据标注和不同解码器损失对行为分类器性能的影响。
- 我们在两个领域中的三个数据集上演示了这些表示，表明我们的方法可以为小鼠减少 10× 的标注，并为果蝇减少 2×。

2. 相关工作

行为建模。使用轨迹数据的行为建模在各个领域都有研究 [26, 6, 39, 42, 20, 3]。特别是，越来越多的人致力于从轨迹数据中自动检测和分类行为 [23, 1, 14, 27, 13, 36]。我们的实验基于行为神经科学的行为分类数据集 [15, 4, 36]，在这个领域，专业领域知识对于识别感兴趣的行为很重要。

行为分析通常包括以下步骤：（1）跟踪智能体的姿态，（2）计算基于姿态的特征，以及（3）训练行为分类器 [4, 21, 36, 28]。为了解决步骤 1 ，有许多现有的姿态估计模型 [15, 27, 18, 36]。在我们的工作中，我们利用了两个现有的姿势模型，分别是小鼠的 [36]和果蝇的 [15]，以生成轨迹数据。在典型行为分析流程的步骤 2 和步骤 3 中，根据动物的姿势计算手工设计的轨迹特征，并训练分类器以完全监督的方式预测感兴趣的行为 [4, 21, 15, 36]。训练完全监督的行为分类器需要领域专家进行耗时的标注 [1]。相反，我们提出的方法使领域专家能够将耗时的标注工作转换为任务编程和表示学习。

另一组工作使用无监督方法来发现新的主题和行为 [22, 41, 2, 26, 5]。我们的工作侧重于更常见的情况，即领域专家已经知道他们想在实验中研究什么类型的动作。我们的目标是提高学习专家定义行为的数据效率。

表示学习。视觉表示学习在图像和视频的有效表示方面取得了很大进展 [17, 16, 7, 29, 25, 19, 38]。自监督信号通常用于训练这种视觉表示，例如学习图

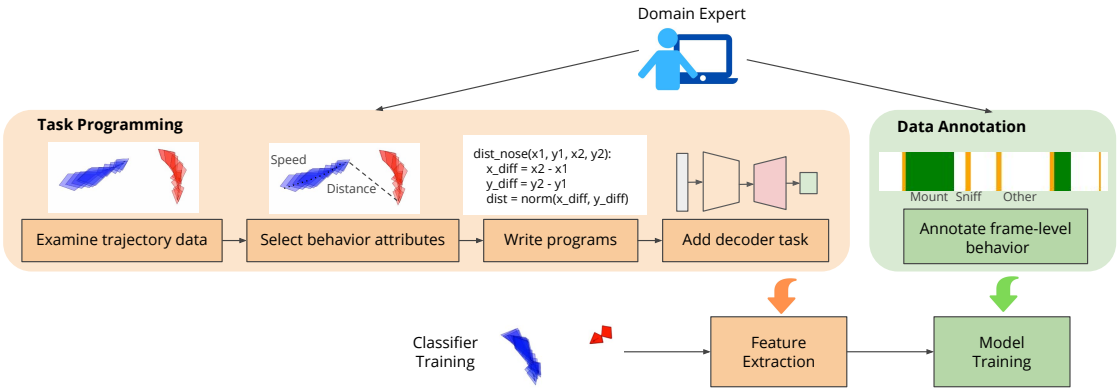

图 2. **用于分类器训练的任务编程和数据标注。**领域专家可以选择进行任务编程或数据标注。任务编程是领域专家为表示学习设计解码器任务的过程。这些程序可以学习标注样本有效的轨迹特征，以提高性能，而不是进行额外的标注。

像块 (patches) 的相对位置 [11]、预测图像旋转 [16]、预测未来的块 (patches) [29]、以及增强图像上的结构学习 [7]。与视觉数据相比，轨迹数据在每一帧中的维数要低得多，并且视觉表示学习的技术通常无法直接应用。例如，虽然我们可以创建表示相同视觉类的图像块，但很难选择表示相同行为的部分关键点集。我们的框架基于这些方法来学习行为数据的有效表示。

为了学习有效的行为表示，我们研究了不同的解码器任务。我们研究的一项解码器任务是自解码：使用生成式建模重建输入轨迹。生成式建模以前已经被应用于学习视觉数据的表示 [45, 38, 29]和语言建模 [31]；对于轨迹数据，我们使用模仿学习 [40, 44, 43] 来训练我们的轨迹表示。我们的多任务自监督学习框架中的其他任务是由领域专家使用任务编程（Section 3.2）创建的。这种使用人类提供的函数作为训练一部分的想法已经被研究用于训练集创建 [33, 32]和可控轨迹生成 [43]。我们的工作探索了这些额外的解码器任务，以进一步改进学习到的表示。

多任务自监督学习。我们在编码器-解码器设置中共同优化了一系列自监督任务，使这项工作成为多任务自监督学习的一个例子。多任务自监督学习已应用于其他领域，例如视觉数据 [12, 25]、加速度计记录 [35]、音频 [34] 和多模态输入 [37, 30]。通常在这些领域中的每一个领域，任务都是提前定义的，例如帧重建、着色、查找图像块的相对位置和视频-音频对齐等任务。大多数这些任务是为图像或视频数据设计的，不能直接应用于轨迹数据。为了构建用于轨迹表示学习的任务，我们建议领域专家可以使用任务编程来设计解码器任务并编码结构知识。

3. 方法

我们引入了 Trajectory Embedding for Behavior Analysis (TREBA)，一种使用领域专家设计的自监督和辅助解码器任务学习标注样本有效轨迹表示的方法。图 2 概述了专家的作用。在我们的框架中，领域专家通过构建少量程序化任务来替代（大量）耗时的手动标注，从而减少了专家的总工作量。每个任务都对学习到的轨迹嵌入施加了额外的约束。

TREBA 使用基于多任务自监督学习方法的专家编程任务，如图 3 所示。为了学习姿态轨迹的任务相关低维表示，我们在（1）输入轨迹的重建（Section 3.1）和（2）专家编程的解码器任务（Section 3.3）上联合训练网络。然后可以将学习到的表示用作行为建模任务的输入，例如行为分类。

3.1. 轨迹表示

设 $\mathcal{D}$ 是 N 个未标记轨迹的集合。每个轨迹 τ 是一个状态序列 $\tau = \{(s_t)\}_{t=1}^{T}$，其中 i 时刻的状态 s_i 对应于智能体在该时刻的位置或姿势。在这项研究中，我们将来自较长记录的轨迹划分为长度为 T 的段，但通常轨迹长度可能会有所不同。对于多个智能体，需要记录每个智能体在每个时刻的关键点。

在介绍我们的专家编程任务之前，我们将使用轨迹重建作为初始的自监督任务。给定代理状态的历史，我们希望我们的模型预测下一个状态。此任

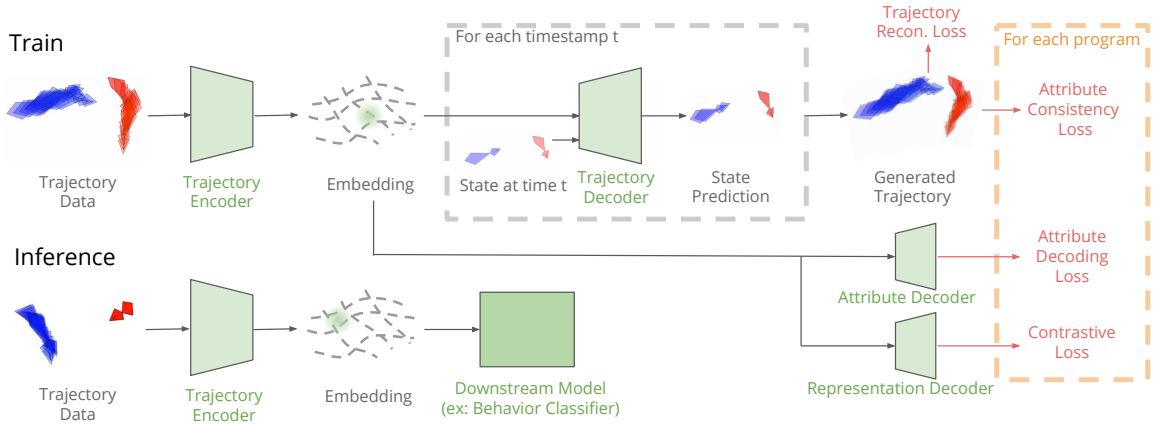

图 3. TREBA **训练和推理流水线**。在训练期间，我们使用轨迹自解码和编程的解码器任务来训练轨迹编码器。学习到的表示用于下游任务，例如行为分类。

务通常使用顺序生成模型进行研究。我们使用轨迹变分自动编码器 (TVAEs) [9, 43] 使用 RNN 编码器 q_ϕ 和 RNN 解码器 p_θ 嵌入输入轨迹来预测下一个状态。TVAE 损失为：

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

我们在 z 上使用先验分布 $p_\theta(\text{z})$ 来正则化学习到的嵌入；在这项研究中，我们的先验是单位高斯分布。通过仅优化 TVAE 损失，我们学习了无监督版本的 TREBA。在执行分类等后续行为建模任务时，我们使用嵌入均值 z_μ。

3.2. 任务编程

任务编程是领域专家为轨迹自监督学习创建解码器任务的过程。这个过程包括从轨迹数据中选择属性、编写程序以及基于程序创建解码器任务（图 2）。在这里，领域专家是具有研究行为的专业知识的人，例如神经科学家或运动分析师。

首先，领域专家从与研究中感兴趣的行为相关的轨迹数据中识别属性。行为属性捕获可能与智能体行为相关的信息，但未明确包含在轨迹状态 $\{(s_t)\}_{t=1}^{T}$ 中。这些属性代表了领域专家在进行行为分析时隐式或显式考虑的结构化知识，例如两个智能体之间的距离、智能体的速度或智能体身体部位的相对位置。

接下来，领域专家编写一个程序来计算轨迹数据的这些属性，这可以使用现有工具完成，例如 MARS [36] 或 SimBA [28]。算法 1 展示了一个来自小鼠社交行为领域的示例程序，用于测量一对正在交互的小鼠之间的“面对角 (Facing Angle)”。每个程序都可以用于构建用于自监督学习的解码器任务（Section 3.3）。

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 (ϕ_1)

$x_{\text{diff}} = x_2 - x_1$

$y_{\text{diff}} = y_2 - y_1$

$\theta = \arctan(y_{\text{diff}}, x_{\text{diff}})$

Return $\theta - \phi_1$

我们的框架受到数据编程范式的启发 [33]，该范式将程序应用于训练集的创建。相比之下，我们的框架使用任务编程将编码结构化专家知识的专家工程程序与表示学习统一起来。

与行为神经科学领域的专家合作，我们创建了一套用于研究我们方法的程序。所选程序是 [36] 中的行为属性子集（对于小鼠数据集）和 [15] 中的行为属性子集（对于果蝇数据集）。我们列出了表 1 中使用的程序，并在补充材料中提供了有关程序的更

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

表 1. **任务编程中使用的行为属性**。在我们的实验中，我们将编程任务建立在每个领域的领域专家提供的这些行为属性的基础上。

多详细信息。

3.3. 学习算法

我们开发了一种方法，将领域专家的程序作为 TREBA 的额外学习信号。我们考虑以下三种方法：(1) 在生成的轨迹 (Section 3.3.1) 中强制执行属性一致性，(2) 直接执行属性解码 (Section 3.3.2)，(3) 基于程序监督应用对比损失 (Section 3.3.3)。这些方法中的每一种都对轨迹 τ 的低维表示 z 应用不同的损失函数。这些解码任务的任何组合都可以与来自 Section 3.1 的自解码相结合，以生成轨迹嵌入 z。

3.3.1 属性一致性

设 λ 是一组 M 领域专家设计的度量智能体行为属性的函数，例如智能体的速度或面对角。回想一下，每个 $\lambda_j, j = 1...M$ 都以轨迹 τ 作为输入，并返回一些专家设计的属性 $\lambda_j(\tau)$ ，该属性是根据该轨迹计算得出的。对于为单个帧设计的 λ_j，我们将该函数应用于 τ 的中心帧。属性一致性旨在为生成的轨迹保持与原始轨迹相同的行为属性标签。设 $\tilde{\tau}$ 为 TVAE 生成的轨迹，给定与 τ 相同的初始条件和编码 z。属性一致性损失为：

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

在这里，我们展示了分类 λ_j 的损失，但一般来说，λ_j 是连续的，并且任何损失度量 $\lambda_j(\tilde{\tau})$ 和 $\lambda_j(\tau)$ 都适用，例如均方误差。我们不要求 λ 总是可微，我们使用 [43] 中引入的可微近似来处理不可微的 λ。

3.3.2 属性解码

另一种选择是直接从学习到的表示 z 中解码每个属性 $\lambda_j(\tau)$。在这里，我们将浅层解码器 f 应用于学习到的表示，解码损失为：

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

类似于方程 (2)，我们展示了分类 λ_j 的损失，但是可以使用任何类型的 λ。

3.3.3 对比损失

最后，编程任务可用于监督表示的对比学习。对于轨迹 τ_i 和每个 λ_j，正例是那些在 λ_j 下具有相同属性类的轨迹。对于具有连续输出的 λ_j，我们创建了一个离散化的 $\hat{\lambda}_j$，其中我们应用固定的阈值将输出空间划分为类。对于我们的工作，我们为每个程序应用两个阈值，以便我们的类大小大致相等。

我们将浅层解码器 g 应用于学习到的表示，并让 $\mathrm{g} = g(q_\phi(\mathrm{z}_\mu|\tau))$ 表示解码后的表示。然后我们应用对比损失：

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

其中 B 是批大小 (Batch Size)，$N_{pos(i,j)}$ 是 τ_i 与 λ_j 的正匹配数，$t > 0$ 是标量温度参数。我们由任务编程监督的对比损失形式类似于 [24] 中由人工标注监督的对比损失。任务编程的一个好处是，与耗时的专家监督相比，程序监督可以快速且可扩展地应用于未标记的数据集。我们注意到，在 [7] 中研究了这种对比损失的无监督版本，其基础是以前的工作，如 [29]。

3.3.4 数据增强

我们可以根据专家提供的程序对轨迹数据进行数据增强。给定所有可能的增强集，我们将 Λ 定义为 *保留属性*的增强集子集：即，对于程序集中的所

有 λ_j，有 $\lambda_j(\tau) = \lambda_j(\Lambda_m(\tau))$，$\Lambda_m \in \Lambda$。小鼠域中有效增强的一个示例是轨迹数据的反射。

通过将损失中的 τ 替换为 $\Lambda_m(\tau)$，可以通过数据增强来扩展上述所有损失。对于对比损失，添加数据增强对应于将批大小扩展到 $2B$，其中 B 个样本来自原始轨迹，其余来自增强轨迹。

我们在实验中使用的增强是反射、旋转、平移和关键点上的小高斯噪声（仅小鼠数据）。在实践中，我们为每个解码器添加了损失，无论是否使用数据增强。

4. 实验

4.1. 数据集

我们使用来自行为神经科学的数据集，其中有来自科学实验的大规模、专家标注的数据集。我们研究了实验室小鼠和果蝇的行为，这两种行为神经科学中最常见的模式生物。对于每个有机体，我们首先使用大型未标注数据集训练 TREBA：对于小鼠域，我们使用一个内部数据集，该数据集由大约 100 小时的二元社交互动记录组成 (Mouse100)，而对于果蝇域，我们使用没有标注的 Fly vs. Fly 数据集 [15]。

在对 TREBA 进行预训练后，我们在三个额外的数据集上评估我们的轨迹表示对监督行为分类（对连续轨迹数据上的帧级行为进行分类）的适用性：

MARS。MARS 数据集 [36] 是最近发布的小鼠社交行为数据集，收集的条件与 Mouse100 相同。该数据集由神经生物学家逐帧标注三种行为：嗅、攻击和攀爬。我们使用其提供的训练、验证和测试集（分别为 781k、352k 和 184k 帧）。轨迹由 MARS 跟踪器 [36] 提取。

CRIM13。CRIM13 [4] 是由专家逐帧手动标注的第二个小鼠社交行为数据集。为了提取轨迹，我们使用了一个版本的 MARS 跟踪器 [36] 在 CRIM13 上的姿势标注上进行了微调。我们选择了一个视频子集，从中可以可靠地检测到 407k、96k 和 142k 帧的训练、验证和测试集的轨迹。我们评估了在 MARS 中研究的相同三种行为（嗅、攻击、攀爬）的分类器性能。

CRIM13 是对在 Mouse100 上训练的 TREBA 鲁棒性的有用测试，因为 CRIM13 中的记录条件（图像分辨率 640×480，帧速率 25Hz，非居中的笼子位置）不同于 Mouse100（图像分辨率 1024×570，帧速率 30Hz，居中的笼子位置）。

Fly vs. Fly (Fly)。我们使用来自 Fly 数据集的 Aggression 和 Courtship 视频 [15]。这些视频记录了一对果蝇之间的互动，并由领域专家逐帧标注它们的社会行为。我们的训练、验证和测试集分别有 1067k、162k、322k 帧。我们使用 [15] 跟踪的轨迹，并评估在完整训练集中具有超过 1000 帧标注的所有行为（猛冲、威胁、争斗、展翅、盘旋、交配）。

4.2. 训练和评估过程

我们使用属性一致性损失（Section 3.3.1）和对比损失（Section 3.3.3）来使用程序训练 TREBA。对于相同的程序，我们发现不同的损失组合会导致相似的性能，一致性和对比损失的组合总体上表现最好。补充材料中提供了所有损失组合的结果。

对于小鼠域（MARS 和 CRIM13） 中的数据集，我们使用小鼠行为领域专家提供的 10 个程序在 Mouse100 上训练 TREBA。对于 Fly 数据集，我们使用果蝇行为领域专家提供的 13 个程序在没有标注的 Fly 训练集上训练 TREBA。完整列表在表 1 中。然后，我们使用经过训练的编码器和预训练的冻结权重，作为 $T = 21$ 帧的轨迹特征提取器，其中每帧的表示是使用当前帧前后的十帧计算的。

我们使用平均精度 (MAP) 来评估我们的分类器，无论有无 TREBA 特征。我们以相同的权重计算感兴趣的行为的平均值。我们的分类器是基于输入特征的浅层全连接神经网络。为了确定分类器性能和训练集大小之间的关系，我们通过随机采样轨迹（长度为 100 帧）对训练数据进行子采样，以达到训练集大小的期望比例。执行采样以实现与完整训练集类似的类分布。对于每个训练比例（1%、2%、5%、10%、25%、50%、75%、100%），我们在三个不同的随机选择的训练数据上训练每个分类器九次。补充材料中提供了其他实施细节。

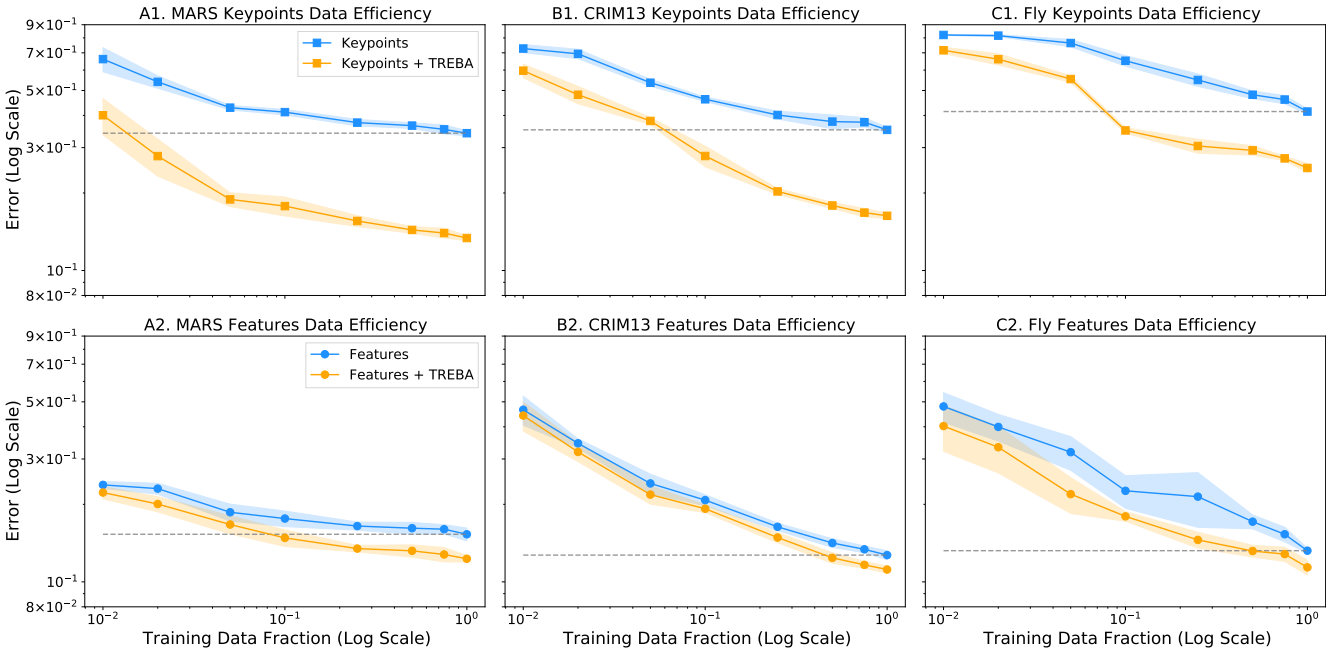

图 4. **监督分类的数据效率。**MARS（左）、CRIM13（中）和 fly（右）上的训练数据分数与分类器误差。蓝线代表基线关键点和特征的性能，橙色线代表 TREBA。阴影区域对应于九次重复的分类器标准偏差。灰色虚线表示在基线特征上进行训练（使用完整训练集）时观察到的最佳分类器性能。注意 x 轴和 y 轴上的对数刻度。

4.3. 主要结果

我们通过训练分类器来预测给定我们学习的表示和 (1) 原始关键点或 (2) 专家设计的特定领域特征之一的行为标签，来评估我们的表示对监督行为分类的数据效率。TREBA+ 关键点评估允许我们在没有其他手工设计特征的情况下测试我们表示的有效性，而 TREBA+ 特征评估更接近大多数潜在用例。小鼠的特定领域特征是来自 [36] 的轨迹特征，果蝇的特征是来自 [4] 的轨迹特征。输入特征是我们在表 1 中使用的程序的超集。

我们的表示能够提高关键点和特定领域特征的数据效率，超过所有评估的训练数据可用性（图 4）。我们在下面讨论每个数据集：

MARS。我们的表示比单独的关键点显着提高了分类性能（图 4 A1）。我们仅使用 1% 和 2% 之间的数据就实现了与完整基线训练相同的性能。虽然这个结果部分是因为我们的表示包含时间信息，但我们也可以观察到 A2 中的数据效率与包含时间特征的特定领域特征相比显着提高。使用 TREBA 的分类器具有与完整基线训练集相同的性能，大约有 5% ∼ 10% 的数据（即 10× ∼ 20× 提高了标注效率）。

CRIM13。我们在 CRIM13 上测试了我们的表示的迁移学习能力，CRIM13 是一个具有与 TREBA 训练集 Mouse100 不同图像属性的数据集。我们的表示使用大约 5% 到 10% 的训练数据中的关键点（图 4 B1）实现了与基线训练相同的性能。对于特定领域的特征，TREBA 使用 50% 的数据标注来获得与完整训练基线相同的性能（即，2× 提高了标注效率）。我们的表示能够推广到同一生物体的不同数据集。

Fly。仅使用关键点时，我们的表示需要 10% 的数据（图 4 C1），而对于特征，我们的表示需要 50% 的数据（图 4 C2) 达到与完全基线训练相同的性能。这相当于 2× 提高了标注效率。

4.4. 模型消融

我们执行以下模型消融以更好地描述我们的方法。在本节中，相对于基线的错误减少百分比是所有训练分数的平均值。其他结果在补充材料中。

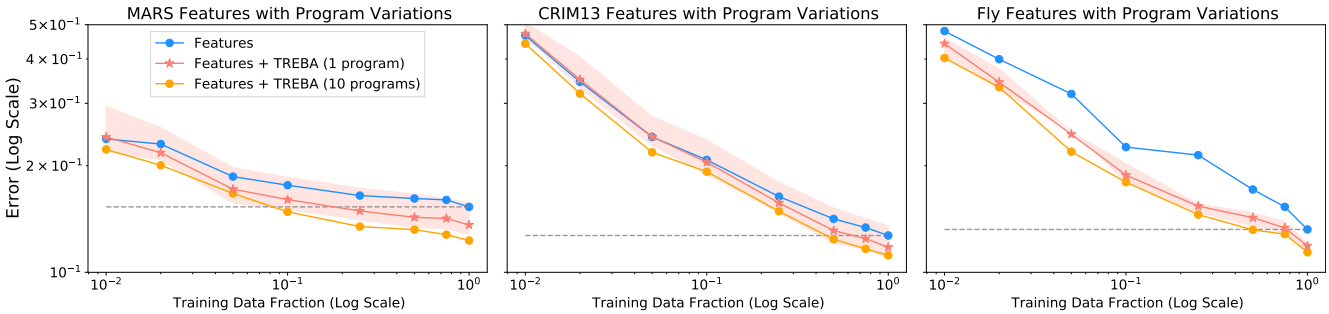

图 5. **不同的编程任务**。不同数量的编程任务对分类器数据效率的影响。阴影区域对应于使用表 1 中的单个编程任务训练的最佳和最差分类器。灰色虚线对应基线特征达到最佳性能的值（使用完整训练集）。

不同的编程任务。我们在表 1 中测试了使用领域专家提供的每个单一程序训练的 TREBA 的性能，图 5展示了平均、最好和最差的性能。平均而言，从单个程序中学习的表示比单独使用特征要好，但使用所有提供的程序可以进一步提高性能。

对于单个程序，根据所选程序的不同，性能可能会有很大差异（图 5）。虽然在分类器 MAP 中性能最好的单个程序接近使用所有程序，但性能最差的程序可能会增加错误，如 MARS 和 CRIM13。我们使用更多程序进一步测试了性能。

在小鼠域中，我们发现与单个程序（补充材料）相比，随机选择三个程序时，它们之间的差异要小得多。使用三个程序，我们实现了从基线特征到使用所有程序的可比平均错误减少（MARS：3 个程序错误减少 14.6%，所有程序错误减少 15.3%，CRIM13：3 个程序减少 9.2%，所有程序减少 9.5%）。对于果蝇域，我们发现我们需要 7 个程序才能达到可比的性能（7 个程序为 20.7%，所有程序为 21.2%）。

不同的解码器损失。当编程任务固定时，一致性（Section 3.3.1）、解码（Section 3.3.2）和对比（Section 3.3.1）损失的不同组合的解码器损失在性能上相似（补充材料）。此外，我们在没有编程任务的情况下评估 TREBA 框架，解码器任务使用轨迹生成和无监督对比损失。虽然自监督表示法在减少基线误差方面也很有效，但我们使用 TREBA 和编程任务（表 2）实现了最佳分类器性能。此外，我们发现，使用来自 [7, 8] 的对比损失，在没有自解码的情况下训练轨迹表示会导致分类表示的效果较差。

数据增强。我们使用 Section 3.3.4 中描述的数据增强去除了损失，发现所有数据集的性能略低于使用增强的性能。特别是，与没有数据增强的情况相比，添加数据增强在 MARS 上减少了 1.2% 的误差，在 CRIM13 上减少了 2.5%，在 Fly 上减少了 5.3%。

	Keypoint Error Reduction (%)		
Decoder Loss	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	**55.1 ± 3.0**	**41.1 ± 2.1**	**33.7 ± 1.2**
	Features Error Reduction (%)		
Decoder Loss	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	**15.3 ± 2.1**	**9.5 ± 3.8**	**21.2 ± 4.5**

表 2. **解码器误差减少**。使用 TREBA 的不同解码器损失进行训练，相对于基线关键点和特定领域特征的误差减少百分比。对所有评估的训练比例取平均值。

改变预训练 MARS 的结果是通过在 Mouse100 上预训练 TREBA 获得的，Mouse100 是一个大型的小鼠内部数据集，具有与 MARS 相同的图像属性。图 6 展示了仅使用 TVAE 和使用程序改变 TREBA 训练数据量的效果。对于关键点和特征，我们观察到 TVAE (MARS) 的误差最大。我们发现，可以通过添加更多数据（特征 + TVAE (Mouse100) 减少 3.9%）或添加任务编程（特征 + 程序（MARS）减少 4.4%）来减少误差。添加更多数据和任务编程可

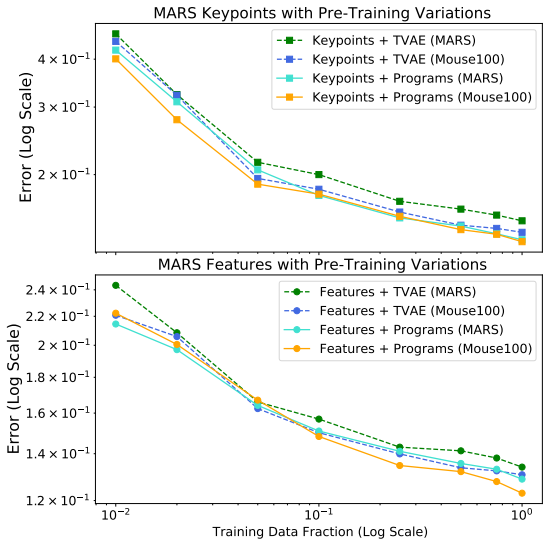

图 6. **改变预训练数据**。不同的预训练数据对 MARS 数据集分类器数据效率的影响。“TVAE”对应于仅使用 TVAE 损失训练 TREBA，而“Programs”对应于所有程序的训练。

使相对于 TVAE (MARS) 的平均误差减少 5.7%，并且平均误差最低。

5. 结论

我们介绍了一种学习标注样本的方法——用于行为分析的有效轨迹嵌入（TREBA）。为了训练这种表示，我们研究了自监督解码器任务以及具有程序监督的解码器任务，后者使用任务编程创建。我们的结果表明，TREBA 可以将小鼠的标注需求减少 10 倍，果蝇的标注需求减少 2 倍。我们在三个数据集（两个在小鼠中，一个在果蝇中）上的实验表明，我们的方法在不同领域都是有效的。TREBA 并不局限于动物行为，还可以应用于跟踪数据标注成本较高的其他领域，如体育分析领域。

我们的实验强调并量化了任务编程和数据标注之间的权衡。哪种方法更有效取决于标注的成本和专家在识别行为属性时的理解水平。创建工具以促进程序创建和数据标注将有助于进一步加速行为研究。

6. 致谢

我们要感谢加州理工学院的 Tomomi Karigo 提供小鼠数据集。西蒙斯基金会 (Global Brain grant 543025 to PP) 慷慨支持这项工作，这项工作得到了 NIH 奖 #K99MH117264 (to AK), NSF 奖 #1918839 (to YY), 和 NSERC 奖 #PGSD3-532647-2019 (to JJS) 的部分支持。

参考文献

[1] David J Anderson and Pietro Perona. Toward a science of computational ethology. Neuron, 84(1):18–31, 2014. 1, 2

[2] Gordon J Berman, Daniel M Choi, William Bialek, and Joshua W Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. Journal of The Royal Society Interface, 11(99):20140672, 2014. 2

[3] Brian Broll, Matthew Hausknecht, Dave Bignell, and Adith Swaminathan. Customizing scripted bots: Sample efficient imitation learning for human-like behavior in minecraft. 1, 2

[4] Xavier P Burgos-Artizzu, Piotr Dollár, Dayu Lin, David J Anderson, and Pietro Perona. Social behavior recognition in continuous video. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 1322–1329. IEEE, 2012. 2, 6, 7

[5] Adam J Calhoun, Jonathan W Pillow, and Mala Murthy. Unsupervised identification of the internal states that shape natural behavior. Nature neuroscience, 22(12):2040–2049, 2019. 2

[6] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8748–8757, 2019. 1, 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. ICML, 2020. 2, 3, 5, 8

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029, 2020. 8

[9] John D Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-consistent trajectory autoencoder: Hierarchical

reinforcement learning with trajectory embeddings. arXiv preprint arXiv:1806.02813, 2018. 4

[10] Anthony I Dell, John A Bender, Kristin Branson, Iain D Couzin, Gonzalo G de Polavieja, Lucas PJJ Noldus, Alfonso Pérez-Escudero, Pietro Perona, Andrew D Straw, Martin Wikelski, et al. Automated image-based tracking and its application in ecology. Trends in ecology & evolution, 29(7):417–428, 2014. 1

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430, 2015. 3

[12] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 2051–2060, 2017. 3

[13] SE Roian Egnor and Kristin Branson. Computational analysis of behavior. Annual review of neuroscience, 39:217–236, 2016. 2

[14] Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. ICLR, 2017. 2

[15] Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. Detecting social actions of fruit flies. In European Conference on Computer Vision, pages 772–787. Springer, 2014. 2, 4, 6

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. ICLR, 2018. 2, 3

[17] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 6391–6400, 2019. 2

[18] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. Elife, 8:e47994, 2019. 2

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019. 2

[20] Katja Hofmann. Minecraft as ai playground and laboratory. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play, pages 1–1, 2019. 1, 2

[21] Weizhe Hong, Ann Kennedy, Xavier P Burgos-Artizzu, Moriel Zelikowsky, Santiago G Navonne, Pietro Perona, and David J Anderson. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. Proceedings of the National Academy of Sciences, 112(38):E5351–E5360, 2015. 2

[22] Alexander I Hsu and Eric A Yttri. B-soid: An open source unsupervised algorithm for discovery of spontaneous behaviors. bioRxiv, page 770271, 2020. 2

[23] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. Nature methods, 10(1):64, 2013. 2

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. arXiv preprint arXiv:2004.11362, 2020. 5

[25] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1920–1929, 2019. 2, 3

[26] Kevin Luxem, Falko Fuhrmann, Johannes Kürsch, Stefan Remy, and Pavol Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. bioRxiv, 2020. 1, 2

[27] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Nature neuroscience, 21(9):1281–1289, 2018. 2

[28] Simon RO Nilsson, Nastacia L Goodwin, Jia J Choong, Sophia Hwang, Hayden R Wright, Zane Norville, Xiaoyu Tong, Dayu Lin, Brandon S Bentzley, Neir Eshel, et al. Simple behavioral analysis (simba): an open source toolkit for computer classification of complex social behaviors in experimental animals. BioRxiv, 2020. 2, 4

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 2, 3, 5

[30] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 133–142, 2020. 3

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 3

[32] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, volume 11, page 269. NIH Public Access, 2017. 3

[33] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In Advances in neural information processing systems, pages 3567–3575, 2016. 2, 3, 4

[34] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6989–6993. IEEE, 2020. 3

[35] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Multi-task self-supervised learning for human activity detection. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(2):1–30, 2019. 3

[36] Cristina Segalin, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J. Sun, Pietro Perona, David J. Anderson, and Ann Kennedy. The mouse action recognition system (mars): a software pipeline for automated analysis of social behaviors in mice. bioRxiv https://doi.org/10.1101/2020.07.26.222299, 2020. 1, 2, 4, 6, 7

[37] Abhinav Shukla, Stavros Petridis, and Maja Pantic. Does visual self-supervision improve learning of speech representations? arXiv preprint arXiv:2005.01400, 2020. 3

[38] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 7464–7473, 2019. 2, 3

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2446–2454, 2020. 1, 2

[40] Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. Robust imitation of diverse behaviors. In Advances in Neural Information Processing Systems, pages 5320–5329, 2017. 3

[41] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. Neuron, 88(6):1121–1135, 2015. 2

[42] Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4610–4619, 2019. 1, 2

[43] Eric Zhan, Albert Tseng, Yisong Yue, Adith Swaminathan, and Matthew Hausknecht. Learning calibratable policies using programmatic style-consistency. ICML, 2020. 1, 3, 4, 5

[44] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. ICLR, 2019. 3

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017. 3