Supplementary - Bar *et al.*, **Sifting through the haystack -** efficiently finding rare animal behaviors in large-scale datasets, *WACV 2025*

Contents

1 Data details

1.1 Synthetic data generation

Each synthetic sample is comprised of a 5-keypoint pose sequence of 9 timesteps. The movement of each keypoint is determined by a sinus with some amplitude A and frequency f, each is drawn from a separate multivariate Gaussian distribution with mean $\hat{A}$ and mean $\hat{f}$ respectively. The behaviors differ in that for the common behavior $\hat{f} > \hat{A}$ and for the rare behavior $\hat{A} > \hat{f}$ (see main text Figure 4). For the Y coordinate, to keep the motion relatively simple, we set $\hat{A}$ to be a small perturbation, varying the movement mostly in the X coordinate of each vertice. This qualitatively made samples appear more similar to the movement in the biological fish datasets we used (see main text Figure 3).

The amplitudes and frequencies of keypoints on the same sample are weakly correlated (covariance = 0.3). For the train set we set $\hat{A}_1 = 0.6$ and $\hat{f}_1 = 6$ for the common behavior and $\hat{A}_2 = 6$ and $\hat{f}_2 = 0.6$ for the rare behavior. For the test set we set $\hat{A}_1 = 0.3$ and $\hat{f}_1 = 3$ for the common behavior and $\hat{A}_2 = 3$ and $\hat{f}_2 = 0.3$ for the rare behavior. We changed the means of the parameters to test whether the model learned the kinematic rule.

We created a separate dataset for each behavior similarity level. We also tested the effect of the baseline rarity level in the dataset on pipeline performance. To do that, we generated datasets with different initial rarity levels - ($\%rarity = 1.5\%, 5\%, 12\%, 24\%$ of the dataset). We created a total of 16 datasets (4 behavior similarities X 4 rarity levels). The number of samples in each dataset varied slightly due to the rarity modulation. All dataset sizes are summarised in Table S1. The code to generate the data is provided in our codebase, and the exact datasets used are provided in the following data repository: 10.5281/zenodo.14266407.

		train			test		
data_type	name	normal	abnormal	total	normal	abnormal	total
synthetic	5% rarity	30000	1500	31500	10000	525	10525
	1.5% rarity	30000	450	30450	10342	158	10500
	12% rarity	30000	3600	33600	9240	1260	10500
	24% rarity	30000	7200	37200	7980	2520	10500
biological	FishLarvae1	111985	15412	127397	22546	3545	26091
	PoseR	25780	1156	26936	4557	202	4759
	Meerkat	58477	3435	61912	11667	716	12383

Table S1: Number of samples for each dataset. Number of normal, abnormal, and total samples for the train and test sets of each dataset. For the synthetic datasets, sample sizes were the same for different similarity levels within the same rarity level.

1.2 Biological datasets - details on datasets and splits

1.2.1 FishLarvae1

Johnson et al. [5] documented larval Zebrafish (*Danio rerio*) chasing prey in a large arena in the laboratory. Videos were shot by tracking the fish across an arena with an overhead camera at 60 frames per second (FPS). Videos were subsequently pose-estimated and then segmented into clips according to the behaviors and thus have varying durations. The pose sequence clips were made publicly available. Behavioral labels were assigned using unsupervised behavioral clustering and divided by researchers into 5 main behavioral categories - explore, pursuit, j-turn, abort, and strike.

The behavior clips were segmented by the annotator such that the first frame is the start of the behavior and the last frame is the end. Such that the behavior instances had varying numbers of timesteps (or frames). This is unlike the two other datasets which have a set duration for all samples. The original dataset included several frames preceding or succeeding the behavior, but we ignored these as they had no behavior annotations.

We note that larval fish striking behavior is very fast ($\sim$ 40 milliseconds), and the framerate this dataset was acquired in is thus sub-optimal to document the behavior. For comparison, the PoseR dataset (below) was acquired at 300 FPS. Indeed, when we reviewed examples from the dataset, the characteristic s-shaped posture preceding a strike appeared only in a single frame. This makes the distinction between behaviors particularly challenging.

Data splitting The data had been acquired in several long filming sessions (trials) and repeatedly from roughly 100 different individual fish. Individual trials may be on different days. Each behavior sequence, i.e., an annotated pose sequence, has an associated trial ID and individual ID. We split the data into test and train such that clips from the same filming trial are all in the same partition (either test or train) however we didn't consider individual identity as each individual had multiple trials. We provide code for data preparation in our codebase which includes the data splits we used.

1.2.2 PoseR dataset

Mullen et al. [7] present another larval Zebrafish behavioral dataset. The data is compiled from several separate neurobiological experimental assays. Unlike the previous dataset, here the motion of the larvae is restricted to a small $25mm\ x\ 25mm\ x\ 25mm$ aquarium. Videos were acquired at 300 FPS and then pose-estimated; these pose sequences were made publicly available. Each behavior sequence was 1 second, i.e., 300 frames. Behavioral labels were assigned using unsupervised behavioral clustering and divided into - burst swim, routine turn, j-turn, scoot, long-latency C-bends, slow-latency C-bends, O-bends, and noise.

Data splitting and cleaning Even though the dataset had a dedicated "noise" category, we found that in many cases skeleton vertices would flicker and be estimated far from the rest of the skeleton. At the same time, we found that our anomaly detector is particularly good at finding these samples and assigning them a high anomaly score. While potentially useful, this was not what we attempted to do in this study. So we filtered the PoseR dataset by dropping frames that had landmarks that were above a threshold distance from the rest of the fish. Since the dataset was acquired at a high frame rate, dropping a single frame did not affect the smoothness of the movement. However, we took a conservative approach, and if a clip had more than 50 frames non-consecutive dropped, or more than 20 consecutive frames dropped we completely removed the clip from the data. In total, we removed 1976 clips from the train data and 344 from the test data using this method.

As for data splitting into train and test, we used the same splits used in the original paper [7]. The cleaning code with the data preparation code is available in our code repository.

1.2.3 Meerkat

We used an accelerometry dataset acquired by Chakravarty et al. [3] to show the generality of the framework to other data modalities and species. This is a dataset of Meerkat behavior assembled by fitting the animals with collars mounted with tri-axial accelerometers while simultaneously monitoring them with video. Acceleration data was acquired at 100 Hz/axis and split into two-second clips resulting in 200 "frames" with 3 data points per frame for each sequence. The behavioral labels were determined from the videos and divided into four behaviors - foraging, vigilance, resting, and running.

Data splitting and preparation As described in the main text, to make the data compatible with the anomaly detection framework we used we had to make each vertice have two channels. We thus took the planar accelerations along (xy, xz, yz). Data preparation code is provided in our codebase. Each behavior segment was associated with the ID of the filmed individual. A total of 10 meerkats were filmed in 11 filming sessions. Though we initially tried to separate individuals, the task proved challenging as it significantly changed the distribution of behaviors in the dataset. Thus we split the dataset randomly into test and train. Given that our goal in the end is to find rare behaviors within a single dataset efficiently, we feel this does not hurt our evaluation. The splits we used are provided in the data preparation code.

2 Statistical analysis methods and results

2.1 Statistical modeling of rarity experiments

To understand the effects of induced rarity level and labeling effort on our method, and robustly compare it to random sampling, we modeled the rarity experiments for each dataset using linear models in R. The code statistical analysis is available in our codebase. The files containing the raw outputs of the experiments are provided in the following data repository: 10.5281/zenodo.14253658

Below we provide a short paragraph describing the main findings of this analysis. When reading the outputs of such models we look at the coefficients each variable is assigned to assess its effect on performance, and at the p-value (p) to assess whether this effect is statistically significant.

Synthetic data We modeled the AuPRC as a function of sampling method (ours or random), rarity level (log-transformed), and labeling effort, including a three-way interaction between these parameters. This interaction term essentially means that different trends may appear in the data with different combinations of these parameters.

We considered each of the 4 behavior SDs separately and created a separate model for each. The results show that both methods are affected by behavior rarity but in different ways. While our method has a weak negative correlation with the frequency of the rare behavior (i.e., performance increases when behavior is rarer, coefficients between -0.18 - 0.03), random sampling performance is positively correlated and with a stronger effect (coefficients between 0.25-0.119). The results show that our method provides more stable performance across the range of rarities and similarity levels.

In Figures 4 and 5 in the main text, we use the statistical models of each dataset to plot the estimated performance given a set labeling budget of 200 samples at different rarity levels. This is done using the 'visreg' package in R [2]. For Table 1 in the main text, we similarly use the model of each dataset to calculate the estimated mean performance across all rarity levels using the 'emmeans' package in R [6].

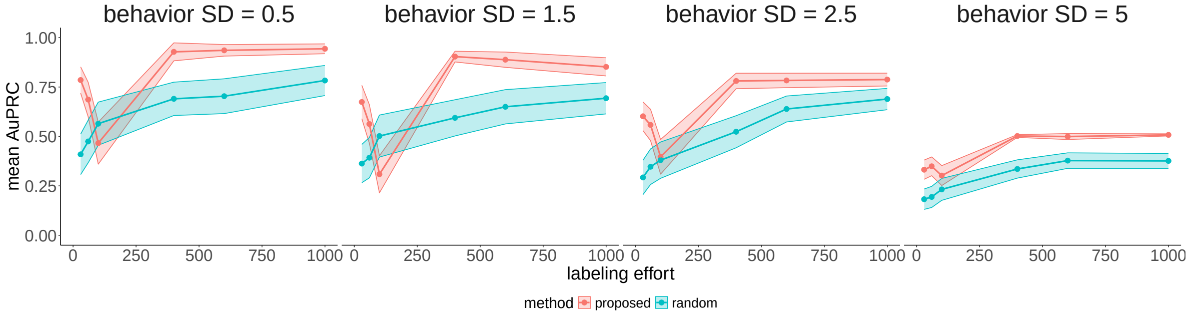

Figure S1: Effect of labeling effort on performance for the **synthetic** datasets. Performance (y-axis) as a function of labeling effort (x-axis) at different behavior similarities (behavior SD, a-d) for our method (red) and the traditional method (blue). Performance was measured in AuPRC and averaged across all tested rarities, the ribbon represents the upper and lower confidence intervals (95% CI). Our method was superior for all labeling efforts, and saturated at around 300 reviewed clips.

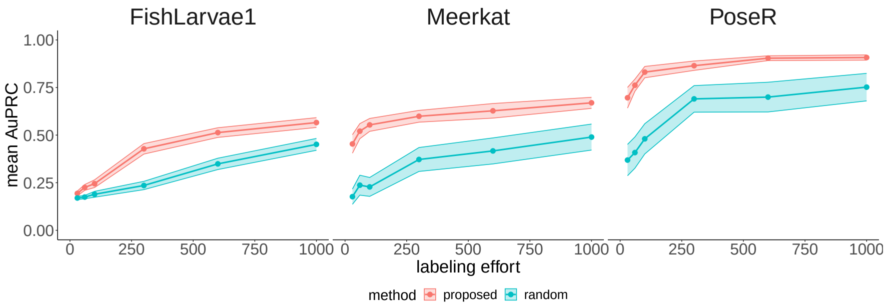

Figure S2: Effect of labeling effort on performance for the **biological** datasets. Performance (y-axis) as a function of labeling effort (x-axis) for the different datasets (a-c) for our method (red) and the traditional method (blue). Performance was measured in AuPRC and averaged across all tested rarities, the ribbon represents the upper and lower confidence intervals (95% CI). Our method was superior for all labeling efforts and saturated after 300 reviewed clips for the FishLarvae1 dataset and only 100 reviewed clips for the PoseR and Meerkat datasets.

2.1.1 Effect of labeling effort

The labeling effort had a small but significant effect on performance for both methods (labeling effort between 30-1000, coefficients for both methods between -0.0004 - 0.0012). In Figures Figure S1 and Figure S2 we plot the actual (i.e., not estimated) mean AuPRCs as a function of labeling effort for the synthetic and biological datasets, respectively. Both figures show similar trends, our method not only yields better performance overall but also does so using a smaller labeling effort. This can be seen by looking at when the curve starts to plateau.

2.2 Statistical analysis of ablation studies

2.2.1 Synthetic data

To statistically evaluate the effect of the behavioral similarity and the frequency of the rare behavior on model performance given the fully labeled dataset (i.e., model benchmarking) we used a linear model. We modeled standardized Area under the Precision-Recall Curve (AuPRC) as a function of behavioral similarity and frequency and included their interaction and the architecture.

Standardized AuPRC The AuPRC expected under a random classifier is equal to the fraction of the positive class in the data. Thus, this metric is sensitive to data imbalance and cannot be used to directly compare performance between test sets with different data imbalances [8, 1]. To standardize the AuPRC we calculate the difference between the AuPRC and the expected performance (which is equal to the %rarity). Because this procedure alone will bias the metric against high behavioral frequencies (with higher expected performance) we further divide the quantity by the maximum possible difference between AuPRC and expected performance 2.2.1. This yields a metric that, like AuPRC, ranges from 0 for poor performance to 1 for perfect performance.

$$standardised\ AuPRC = \frac{AuPRC - \%\,rarity}{1 - \%\,rarity} \tag{1}$$

ST-GCN-based classifiers showed the best performance across the entire range of behavioral frequencies and behavioral similarities (see Figure S3 and Table S2). Surprisingly, if the behavior of interest is quite distinct (behavior SD = 0.5), even high rarity levels (translating into high data imbalance) only mildly affect performance (-0.02 standardized AuPRC for the lowest overlap level). A linear model predicting the standardized AuPRC as a function of the baseline behavioral frequency (rarity), behavioral similarity, and architecture found quite intuitive results. The unsupervised architecture was worse than the supervised one (coefficient = -0.19, $p < 5.8e-13$). The similarity between behaviors (behavior SD) hurts performance (coefficient =-0.15, $p < 0.0001$). The baseline rarity by itself had no significant effect, however, it positively interacts with behavior similarity (coefficient = 0.27, p < 0.0001) which means that when a behavior is more frequent, behavior similarity has less effect on performance. The linear model explained a substantial part of the variance in the data ($R^2 = 0.81$).

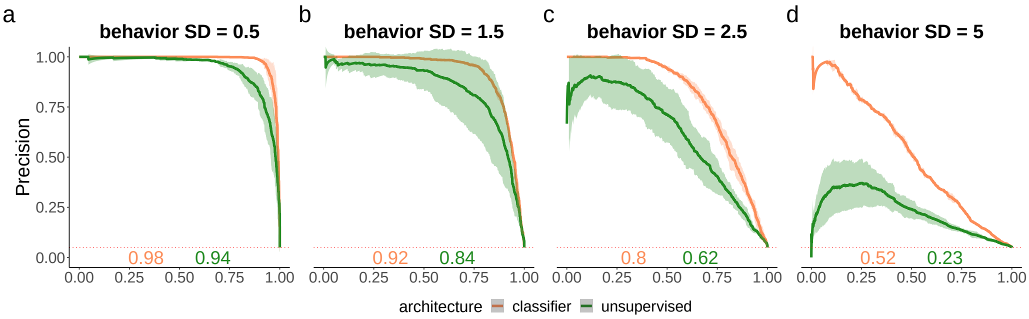

Figure S3: Example of architecture performance on **synthetic data**. Precision-Recall curves for the 5 % behavior rarity dataset for different behavioral overlaps (panels a-d). Colored numbers in each pane correspond to the respective area under each of the curves. ST-GCN classifiers (orange) dominate performance. Unsupervised Normalizing Flows (green) are competitive for lowest behavioral similarities but their performance degrades more quickly.

		Behavior Standard Deviation (more overlap →)			
Method	% Rare	0.5	1.5	2.5	5
Anomaly detector (STG-NF)	1.5%	0.85±0.14	0.56±0.44	0.45±0.1	0.17±0.1
	5%	0.94±0.016	0.83±0.097	0.6±0.096	0.19±0.051
	12%	0.92±0.018	0.67±0.18	0.74±0.12	0.36±0.085
	24%	0.89±0.051	0.79±0.002	0.59±0.027	0.37±0.1
Classifier (ST-GCN)	1.5%	0.97±0.011	0.89±0.014	0.61±0.0023	0.3±0.007
	5%	0.98±0.0077	0.92±0.0038	0.79±0.016	0.5±0.0032
	12%	0.99±0.0049	0.96±0.0056	0.88±0.013	0.6±0.0075
	24%	0.99±0.0041	0.98±0.0016	0.92±0.0035	0.71±0.003

Table S2: Architecture performance on **synthetic data**. The standardized area under the precision-recall curve of the two models for different levels of frequency for the rare behavior (rows, column 2) and similarity between the frequent and rare behaviors (columns 3-7). Performance ranges from poor (0.17, dark blues) to excellent (0.99, yellows). ST-GCN classifiers are superior across the board. When behavior is distinct (column 3) anomaly detection yields decent results with no labeling effort invested, however, performance degrades quickly with similarity.

All in all these results establish that, given sufficient data, graph classifiers deliver high performance even under extreme imbalances. Additionally, they highlight that while rarity may not be an issue by itself when dealing with fine-grained behaviors it hurts performance considerably. However, the question that remains is how we find the labeled instances of rare behaviors to train these highly performant classifiers. This motivated us to find a way to quickly obtain and annotate rare behaviors using the anomaly detector which, though less performant, requires no labeled samples.

2.2.2 Biological datasets

In all biological datasets, like the synthetic data, ST-GCNs show superior results (Figure S4) despite the high data imbalance. Unsupervised STG-NFs, for 2 out of 3 datasets, show drastically lower performance. It has been previously shown that STG-NF is adversly affected when the train set has a high percentage of abnormal samples [4]. During some preliminary experimentation, we found this to be partially, though not entirely, the case here. Integrating insights from the synthetic regime, this reduced performance in the unsupervised approach implies a higher degree of similarity between rare and common behaviors.

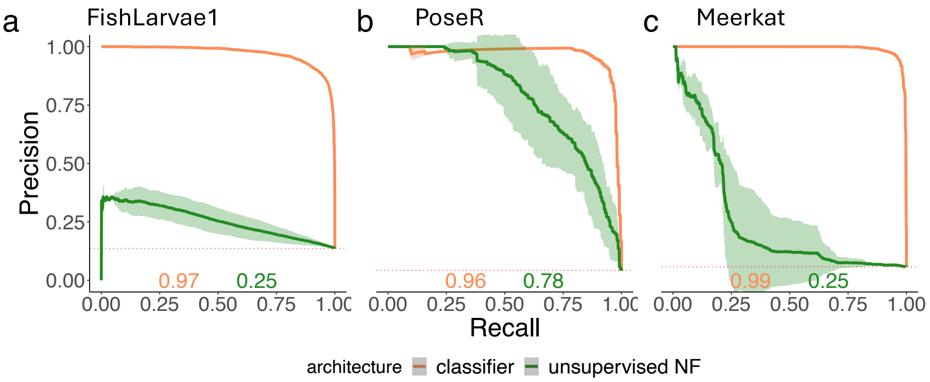

Figure S4: Architecture performance on **biological data**. Precision-Recall curves for each of the experimental datasets (panels a-c). ST-GCN classifiers (orange) dominate performance. Unsupervised Normalizing Flows (green) are not competitive on their own (except for panel b).

References

[1] Jan Brabec, Tomáš Komárek, Vojtěch Franc, and Lukáš Machlica. On model evaluation under non-constant class imbalance. In *International Conference on Computational Science*, pages 74–87. Springer, 2020. 4

[2] Patrick Breheny and Woodrow Burchett. Visualization of regression models using visreg. *The R Journal*, 9(2):56–71, 2017. 3

[3] Pritish Chakravarty, Gabriele Cozzi, Arpat Ozgul, and Kamiar Aminian. A novel biomechanical approach for animal behaviour recognition using accelerometers. *Methods in Ecology and Evolution*, 10(6):802–814, 2019. 2

[4] Or Hirschorn and Shai Avidan. Normalizing flows for human pose anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13545–13554, 2023. 6

[5] Robert Evan Johnson, Scott Linderman, Thomas Panier, Caroline Lei Wee, Erin Song, Kristian Joseph Herrera, Andrew Miller, and Florian Engert. Probabilistic models of larval zebrafish behavior reveal structure on many scales. *Current Biology*, 30(1):70–82, 2020. 2

[6] Russell V. Lenth. *emmeans: Estimated Marginal Means, aka Least-Squares Means*, 2024. R package version 1.10.2. 3

[7] Pierce N Mullen, Beatrice Bowlby, Holly C Armstrong, and Maarten F Zwart. Poser-a deep learning toolbox for decoding animal behavior. *BioRxiv*, pages 2023–04, 2023. 2

[8] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015. 4