Evaluating Long-Tailed Learning Techniques on Pedestrian Trajectory Prediction

Divya Thuremella
Department of Engineering
Oxford University
Oxford, UK
divya@robots.ox.ac.uk

Lars Kunze
Department of Engineering
Oxford University
Oxford, UK
lars@robots.ox.ac.uk

***Abstract*—Autonomous robots have the biggest potential for risk because they operate in open-ended environments where humans interact in complex, diverse ways. To operate, such systems must predict this behaviour, especially it's part of the unexpected and potentially dangerous long tail of the dataset. Since previous work on long-tailed prediction is limited, and uses variably defined long-tailed metrics, we aim to unify the different long-tailed trajectory prediction approaches by comparing them on the same long-tailed metrics and test a new long-tailed learning technique previously not yet applied to trajectory prediction. Furthermore, in order to more fairly compare methods, we advocate for metrics which value multimodal predictions while penalizing random guessing, which is not something that the popular 'best-of-20' metric accomplishes. To our knowledge, we are the first work to compare long-tailed trajectory prediction techniques on metrics which are more practical to autonomous robots, and one of the first to apply long-tailed learning techniques to methods which assign likelihoods to predictions, a feature that is essential for using these predictions in a practical way within autonomous systems.**

***Index Terms*—long-tailed learning, trajectory prediction**

I. Introduction

Most previous studies of long-tailed learning within prediction demonstrate their results by showing an improvement on the Trajectron++EWTA model [1]. However, this model outputs an unranked set of N (typically, 20) future trajectories without ranking them by likelihood. Such models are difficult to use in practice because applications such as autonomous robots will need to plan a course of action for each trajectory that is predicted. Therefore, we propose to evaluate how well these long-tailed learning techniques developed for trajectory prediction perform on methods like Trajectron++ [2], which yield a distribution of possible future trajectories with a likelihood attached to each one. Such methods allow applications to plan for only the most likely futures and are therefore more useful in practice.

'Best-of-20' Metric. Since one history sequence could yield multiple plausible paths (e.g. person approaching an obstacle can go around on the right or left without giving prior indication) such that the ground truth represents only one out of many potentially likely paths, many methods evaluate their models using the multimodal evaluation metrics 'Best-of-N' or 'top-N', where the N (typically, 20) most likely paths are predicted, and the path which is closest to the ground truth is compared to the ground truth to calculate errors [3].

'Most Likely' Metric compares the average distance error (ADE) and final distance error (FDE) between the ground truth path and the single most likely prediction output by the model.

Long-Tailed Learning. We use the term long-tailed to describe most naturally sampled datasets that contain many examples of a few common cases and few examples of many uncommon cases. The uncommon examples in the long tail are harder to predict, as they are rare and dispersed among the many majority cases. Within prediction, there are many examples of easily predictable behaviors like standing still or traveling at a constant velocity, and few examples of complicated behaviors like stopping to tie a shoelace, which makes them harder to identify and predict.

In this work, we 1) unify different long-tailed learning approaches within trajectory prediction which were done in parallel and evaluated on separately defined long-tail metrics, and 2) evaluate whether the good results that these approaches achieve (when applied to models which use EWTA loss [1] with 'Best-of-20' metrics) can be replicated with models which attach likelihoods to their predictions (like Trajectron++ [2]) on the 'Most Likely' metric.

II. Related Work.

A. Long-Tailed Learning in Trajectory Prediction

Within trajectory prediction, [1], [5], and [7] are the only methods, to our knowledge, which directly address long-tailed learning. [1]and [7] use contrastive loss on implicit classes of trajectories to force the model to learn the characteristics of rare trajectories separately from common trajectories. This loss forces the feature embeddings of the rare trajectories away from that of common trajectories [1]. Therefore, feature embeddings of rare trajectories are less likely to be lost within the manifold of common trajectories, and assumed to be outliers. In [1], classes are defined by how easy it is to predict the future trajectory through a physics-based Kalman filter: rare and important trajectories are assumed to be the ones which are difficult to predict using simple kinematics. In [7], unsupervised clustering is performed via an autoencoder to assign examples to pseudo-classes, and class frequency is used to separate common from rare classes. [7] additionally employ

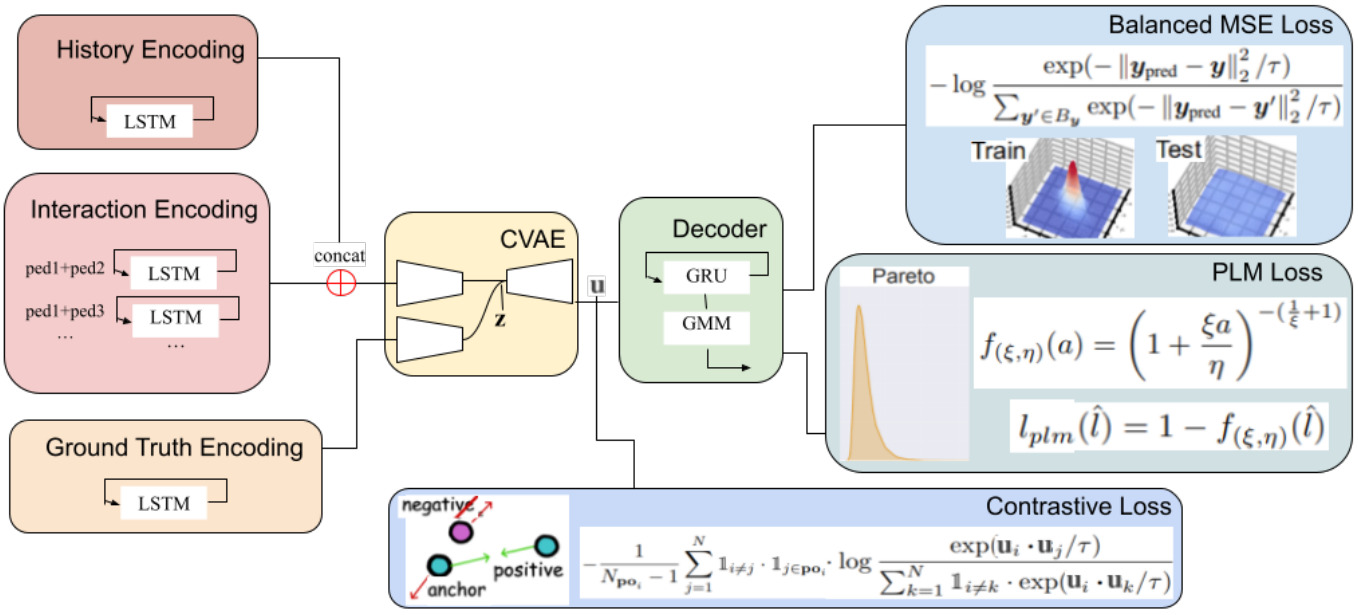

Fig. 1. Architecture of Baseline Model and Application of 3 Long-Tailed Learning Techniques (i.e. Contrastive Loss [1], Balanced MSE Loss [4], and PLM Loss [5]). Contrastive loss pushes the embeddings of nodes in the same class (i.e. similar 'difficulty' level) together, and those of different classes apart, as shown (where τ is a pre-defined hyperparameter and $\mathbf{po}_i$ is the positive set of anchor i, i.e. the set of samples j in the batch which has a difficulty score s_j satisfying $|s_i - s_j| < \theta_p$, where θ_p is a hyper-parameter defining the positivity threshold). Balanced MSE loss pulls predictions towards their ground truth and away from the ground truths of other examples in the batch, because of the assumption that the training set is imbalanced, while the test set is balanced across the output space (as shown in the equation, where τ is a hyperparameter learned during training and B_y is the labels of a batch). PLM loss takes the initially calculated per-example loss, $\hat{l}$, and uses the assumption that the long tail is shaped like a pareto curve to transform it according to the equation shown (where ξ and η are pre-defined hyperparameters). The diagrams of the baseline model architecture are based on [2], while pictures and equations of the contrastive, balanced MSE, and PLM losses are taken from [1], [4], and [5] respectively. The diagram for contrastive loss is from [6].

Hypernetworks to learn different weights for common and uncommon examples. Meanwhile, [5] propose novel losses, of which the best performing is a regularization term which assumes a fixed shape (pareto distribution) for the error. While [1], [5], and [7] show small improvements in averaged metrics, it is difficult to compare improvements in the long tail as each paper defines and optimizes for their own scale of uncommonness. Therefore, we implement [1] and [5] on the Trajectron++ model [2] (since [7] came out too recently without code, it was difficult to implement quickly) and compare their performances on both scales of uncommonness. One caveat: while [7] evaluate their method on the Trajectron++ model, and report improved NLL metrics, they do not compare theirs to other methods on this model/metric.

B. Long-Tailed Learning in Regression

Outside of prediction, there are few works on multi-dimensional regression that incorporate different long-tailed learning techniques (e.g. [4], [8]), but the most applicable to trajectory prediction is balanced MSE [4]. [4] operates on the assumption that the test set is balanced across the output trajectory space, even if the training set is imbalanced and/or long-tailed. We use balanced MSE [4] as an additional loss term and compare performance to that of other methods.

C. Metrics.

Most methods that use the ETH-UCY dataset, to our knowledge, use the best-of-20 metric but this allows high scores to be achieved by 'shot gun' predictions (i.e. evenly spread uninformed guesses) [9], and isn't a useful indication of how well these methods will perform in applications where each possible future needs to be planned for. Therefore, more useful metrics which still promote multimodality include: Negative Log-Likelihood over the predicted distribution of future trajectories [9], and using a subset of most likely modes instead of 'best' mode [10]. However, such metrics require models that assign likelihoods to each predicted trajectory, which Trajectron++EWTA [1] doesn't do. [3] propose using 'best-of-3' instead of 20 for such methods which can't assign likelihoods, but this is not used in the long-tailed learning prediction literature.

III. Methodology

In order to unify the different long-tail learning techniques applicable to trajectory prediction, as well as test their performances on a model capable of producing trajectories with likelihoods, we implemented each of these techniques on Trajectron++ [2] baseline model, and evaluated both the most likely and best-of-20 (previously reported metric) scores of each method on both the long-tail metrics defined by [1] and [5].

The long-tailed learning techniques within trajectory prediction that we evaluated are: 1) Contrastive Loss proposed by [1], 2) PLM Loss, the best of the long-tailed learning techniques proposed by [5], and 3) balanced MSE loss proposed by [4], which was developed for regression tasks but has not yet been applied to trajectory prediction. A diagram summarizing these three additional losses and how they were incorporated into the architecture of the baseline model is shown in Figure 1.

A. Implementation of Models

The baseline model we use to compare long-tailed learning methods is Trajectron++ [2], as it produces a distribution of future trajectories and their likelihoods, which is useful in planning applications. In contrast, past long-tailed trajectory prediction methods ([1] and [5]) have used the Trajectron++EWTA model [1] as a baseline since its 'Best-of-20' ADE/FDE metrics show better performance than Trajectron++ [2]. However, this comes at a cost: the EWTA (Evolving Winner-Takes-All) loss always predicts N (in this case, 20) future trajectories without any associated likelihoods, and specifically optimizes for the 'Best-of-20' metric by 'evolving' the training scheme such that in the beginning, loss is averaged across all 20 trajectories, but by the end of the training, loss is only optimized for the single trajectory that is closest to the ground truth [11].

To train our baseline model, we maintain the same training methodology and parameters as [2]. The metrics reported in this work are those achieved by our retrained iteration of [2], and are consistent with the metrics reported in [2].

Contrastive Loss. We implemented the Contrastive loss proposed by [1] by taking the vector $\boldsymbol{u}$, shown in Figure 1, and using it as the contrasted feature embedding. All other parameters of the contrastive loss were taken from the default values in [1]. We combined this loss with the baseline loss in ratios of 1:1 and 1:10 (original : contrastive loss, respectively), and used a batch-size of 256.

PLM Loss. We use the PLM loss [5], by applying the function shown in Figure 1 to the individual loss of each example, and adding their average to the baseline loss in varying degrees, as shown in Table I. A ratio of p indicates that the loss is p% PLM loss and (1 - p)% baseline loss. After a hyper parameter search, we found best results using $\eta = 100$ and an $\xi = 0.01$. We also multiplied the PLM loss by 100 to make it the same order of magnitude as the baseline loss.

Balanced MSE Loss. We use the Batch-based Monte Carlo implementation of the re-balancing loss proposed by [4] which uses cross-entropy to pull each example closer to its ground truth and further from the ground truth of other examples in the batch. Ratios are calculated in the same way as PLM loss. Although the assumption by [4] that the test set is balanced, does not fully apply, using this assumption forces the model to treat the rare examples as if they are just as common as the most frequently seen examples.

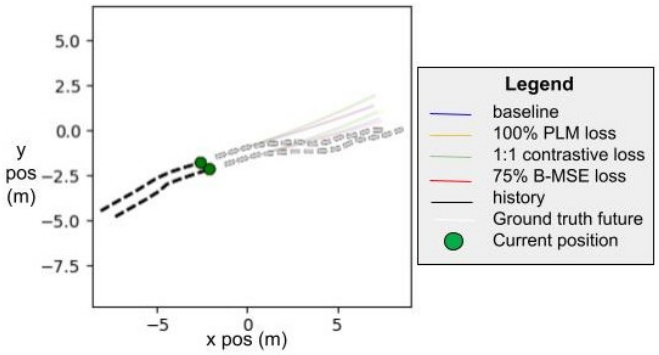

Fig. 2. Representative instance of examples within the worst performing 1% of the dataset on the baseline model. As can be seen, even in cases which fall into the worst performing 1% on the baseline model, adding each of the three losses doesn't make much difference.

IV. Results and Conclusions

As can be seen by the results in Table I, some of the methods improve the errors slightly on some datasets, but no method achieves significantly better results than the baseline.

To unify the results of [1] and [5] we have evaluated all three long-tailed learning methods on both the long-tailed metric proposed by [1] (shown in Table II) and the long-tailed metric proposed by [5] (shown in Table III). While [1] uses their 'difficulty scoring' to get the ADE/FDE of the most difficult 1, 2, and 3 percent of examples, [5] calculates the 95th, 98th, and 99th percentile of the distribution of errors to measure long-tail performance. Both of these methods, however, only calculate these long-tailed metrics on the Best-of-20 prediction. Therefore, we additionally calculate both long-tailed trajectory prediction metrics on the most likely single trajectory predictions.

From Tables II and III, it can be seen that the two long-tail metrics are consistent: the best performing methods on the 99th and 98th percentile Most Likely metrics are the same methods which perform best on the Top 1 and Top 2 Most Likely prediction metrics, respectively. However, the Most Likely metrics and Best-of-20 metrics are not correlated: methods which perform the best on each of the two metrics for a specific dataset do not correspond, shown in Table I.

Moreover, all 3 methods perform fairly similarly to the baseline on all the metrics presented, as shown in Figure 2. This may be because all of these methods are re-balancing methods [12], a type of long-tailed learning technique which uses the loss to force the model to treat all examples equally. Although class re-balancing methods are generally simple to implement and show minor performance improvements, they essentially can't handle the issue of lacking information due to limited data so improving tail performance typically involves the trade-off of also regressing head performance to some extent [12]. Furthermore, the heuristic division of the dataset into rare and frequent sets causes a tendency to classify outliers as important, rare examples [13].

TABLE I
PER-DATASET PERFORMANCE ACROSS TEST SET FOR MOST LIKELY AND BEST-OF-20 PREDICTIONS

Method	Ratio	Most Likely										Best-of-20									
		ETH		HOTEL		UNIV		ZARA1		ZARA2		ETH		HOTEL		UNIV		ZARA1		ZARA2	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Baseline	100%	**0.70**	1.68	0.22	0.46	0.41	1.06	0.30	0.77	0.22	0.57	0.42	0.85	**0.12**	0.20	0.22	0.43	0.17	**0.31**	0.12	**0.24**
B-MSE	25%	0.71	1.69	0.22	0.47	0.40	1.04	**0.29**	0.76	0.22	0.57	0.43	0.89	**0.12**	**0.19**	**0.21**	**0.42**	0.17	**0.31**	0.12	**0.24**
B-MSE	50%	0.71	1.71	0.22	0.47	0.39	1.04	0.30	0.77	0.22	0.57	0.43	0.87	**0.12**	**0.19**	0.22	**0.42**	0.17	0.32	0.12	0.25
B-MSE	75%	0.71	1.70	0.22	0.47	0.39	1.01	**0.29**	**0.75**	0.22	0.57	0.42	0.85	**0.12**	**0.19**	0.22	**0.42**	**0.16**	**0.31**	0.12	0.25
Contr.	1:1	0.72	1.71	0.22	0.47	0.47	1.24	0.30	0.77	0.22	0.57	0.44	0.87	0.13	0.21	**0.21**	**0.42**	0.17	0.32	0.12	0.25
Contr.	10:1	**0.70**	**1.67**	0.22	0.47	0.42	1.15	**0.29**	0.76	0.22	0.57	**0.41**	**0.79**	0.13	0.22	**0.21**	**0.42**	**0.16**	0.32	0.12	0.25
PLM	25%	0.71	1.70	**0.21**	0.46	0.40	1.04	**0.29**	**0.75**	0.22	0.57	0.43	0.87	**0.12**	**0.19**	**0.21**	0.43	**0.16**	0.32	0.12	0.25
PLM	50%	**0.70**	**1.67**	**0.21**	0.46	0.39	1.01	**0.29**	0.76	0.22	0.57	0.43	0.87	**0.12**	**0.19**	**0.21**	**0.42**	0.17	0.32	0.12	0.25
PLM	75%	0.71	1.71	**0.21**	0.46	0.39	1.01	**0.29**	**0.75**	0.22	0.57	0.44	0.89	**0.12**	**0.19**	**0.21**	**0.42**	0.17	0.32	0.12	0.25
PLM	100%	0.71	1.71	**0.21**	**0.45**	**0.38**	**0.99**	0.30	0.77	0.22	0.57	0.43	0.88	**0.12**	**0.19**	0.22	**0.42**	0.17	0.32	0.12	0.25

TABLE II
MOST LIKELY (ML) AND BEST-OF-20 (BO) AVERAGE PERFORMANCE OF ALL, TOP 1%, 2%, AND 3% MOST 'DIFFICULT' EXAMPLES AS DICTATED BY [1], AVERAGED OVER THE 5 DATASETS (ETH, HOTEL, UNIV, ZARA1, ZARA2)

Method	Ratio	All (ML)		Top 1		Top 2		Top 3		All (BO)		Top 1		Top 2		Top 3	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Baseline	100%	0.37	0.91	1.02	2.31	1.02	2.39	0.97	2.29	0.21	0.41	0.53	1.04	0.56	1.14	0.54	1.12
Balanced MSE	25%	0.37	0.91	1.05	2.33	1.05	2.41	0.98	2.32	0.21	0.41	0.55	1.11	0.58	1.20	0.57	1.18
Balanced MSE	50%	0.37	0.91	1.04	2.37	1.05	2.45	0.98	2.34	0.21	0.41	**0.49**	0.97	0.55	1.10	0.55	1.12
Balanced MSE	75%	0.37	0.90	**0.97**	**2.16**	1.01	**2.33**	0.96	2.26	0.21	**0.40**	0.50	**0.88**	0.58	1.11	0.55	1.09
Contrastive	1:1	0.39	0.95	1.03	2.29	1.03	2.36	0.97	2.28	0.21	0.41	0.55	1.05	0.60	1.17	0.58	1.17
Contrastive	10:1	0.37	0.92	0.99	2.25	**1.00**	**2.33**	**0.95**	**2.25**	0.21	**0.40**	0.53	0.90	0.54	1.02	**0.53**	**1.03**
PLM	25%	0.37	0.90	1.05	2.37	1.04	2.42	0.98	2.32	0.21	0.41	0.50	0.94	0.56	1.16	0.55	1.17
PLM	50%	**0.36**	**0.89**	1.04	2.32	1.03	2.39	0.97	2.29	0.21	0.41	0.57	1.13	0.58	1.17	0.57	1.17
PLM	75%	**0.36**	0.90	1.04	2.35	1.03	2.40	0.97	2.31	0.21	0.41	0.51	0.98	0.57	1.17	0.57	1.22
PLM	100%	**0.36**	0.90	1.02	2.29	1.03	2.37	0.97	2.29	0.21	0.41	0.50	0.91	**0.52**	**1.01**	**0.53**	1.07

TABLE III
MOST LIKELY (ML) AND BEST-OF-20 (BO) MEAN, 95TH, 98TH, AND 99TH PERCENTILE OF ADE AND FDE (METRIC USED BY [5]), AVERAGED OVER THE 5 DATASETS (ETH, HOTEL, UNIV, ZARA1, ZARA2)

Method	Ratio	Avg (ML)		95^{th}		98^{th}		99^{th}		Avg (BO)		95^{th}		98^{th}		99^{th}	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Baseline	100%	0.37	0.91	1.00	2.56	1.26	3.27	1.50	3.73	0.21	0.41	0.56	1.36	**0.76**	**1.88**	0.95	**2.36**
Balanced MSE	25%	0.37	0.91	1.00	2.56	1.27	3.26	1.49	3.73	0.21	0.41	0.57	1.40	0.79	2.01	0.96	2.38
Balanced MSE	50%	0.37	0.91	1.00	2.58	1.27	3.30	1.50	3.73	0.21	0.41	0.58	1.44	0.82	2.08	0.99	2.54
Balanced MSE	75%	0.37	0.90	1.00	**2.52**	1.27	**3.24**	**1.46**	3.73	0.21	**0.40**	0.56	1.34	0.79	2.00	**0.94**	2.37
Contrastive	1:1	0.39	0.95	1.01	2.61	1.26	3.30	1.49	3.74	0.21	0.41	0.57	1.38	0.82	2.00	0.99	2.44
Contrastive	10:1	0.37	0.92	**0.98**	2.54	1.27	**3.24**	**1.46**	**3.70**	0.21	**0.40**	**0.55**	**1.27**	0.79	1.98	0.99	2.41
PLM	25%	0.37	0.90	0.99	2.56	1.26	3.28	1.47	3.74	0.21	0.41	0.57	1.39	0.79	1.95	0.97	2.45
PLM	50%	**0.36**	**0.89**	0.99	2.53	**1.24**	3.26	1.49	3.71	0.21	0.41	0.57	1.40	0.80	1.98	0.98	2.46
PLM	75%	**0.36**	0.90	1.01	2.55	1.26	3.29	1.50	3.71	0.21	0.41	0.58	1.39	0.79	1.99	0.98	2.53
PLM	100%	**0.36**	0.90	1.00	2.57	1.27	3.29	1.49	3.74	0.21	0.41	0.58	1.41	0.82	2.08	0.99	2.52

V. FUTURE WORK

Therefore, our next aim is to use ensemble learning to train a series of experts to recognize and perform well on certain aspects of the dataset (i.e. one expert for similarly behaving common examples, and multiple others for uncommon examples which behave in their own unique ways). Instead of grouping all uncommon examples together, as [1] and [5] do, this will allow the network to learn different types of behaviors corresponding to the different reasons an example may fall into the 'uncommon' category. Furthermore, we intend to incorporate map information and social interaction modeling into our ensemble learning model to add information that will help inform a classifier to distinguish between the possible reasons an example may be performing exceptionally poorly.

REFERENCES

[1] O. Makansi, O. Cicek, Y. Marrakchi, and T. Brox, "On Exposing the Challenging Long Tail in Future Prediction of Traffic Actors," *arXiv:2103.12474 [cs]*, Aug. 2021.

[2] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data," *arXiv:2001.03093 [cs]*, Jan. 2021.

[3] P. Kothari, S. Kreiss, and A. Alahi, "Human Trajectory Forecasting in Crowds: A Deep Learning Perspective," Jan. 2021.
[4] J. Ren, M. Zhang, C. Yu, and Z. Liu, "Balanced MSE for Imbalanced Visual Regression," *arXiv:2203.16427 [cs]*, Mar. 2022.
[5] J. Kozerawski, M. Sharan, and R. Yu, "Taming the Long Tail of Deep Probabilistic Forecasting," *arXiv:2202.13418 [cs]*, Mar. 2022.
[6] "Deep Metric Learning for Signature Verification," https://blog.fastforwardlabs.com/2021/06/09/deep-metric-learning-for-signature-verification.html.
[7] Y. Wang, P. Zhang, L. Bai, and J. Xue, "FEND: A Future Enhanced Distribution-Aware Contrastive Learning Framework for Long-tail Trajectory Prediction," Mar. 2023.
[8] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi, "Delving into Deep Imbalanced Regression," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 11 842–11 851.
[9] E. Pajouheshgar and C. H. Lampert, "Back to square one: Probabilistic trajectory forecasting without bells and whistles," Dec. 2018.
[10] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal Behavior Prediction using Trajectory Sets," Apr. 2020.
[11] O. Makansi, E. Ilg, O. Cicek, and T. Brox, "Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7144–7153.
[12] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep Long-Tailed Learning: A Survey," *arXiv:2110.04596 [cs]*, Oct. 2021.
[13] N. Moniz, P. Branco, and L. Torgo, "Evaluation of Ensemble Methods in Imbalanced Regression Tasks," in *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. PMLR, Oct. 2017, pp. 129–140.