Sensor signal processing

Semisupervised Learning for Noise Suppression Using Deep Reinforcement Learning of Contrastive Features

Ehsan Kazemi[1], Fariborz Taherkhani[2], and Liqiang Wang[1]
[1]Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA
[2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Manuscript received 17 March 2023; accepted 31 March 2023. Date of publication 6 April 2023; date of current version 17 April 2023.

Abstract—In this letter, we present DeDRLSSL: a generic semisupervised noise suppression framework. The proposed model is based on a reinforcement learning system for learning contrastive features to refine the features utilized in consistency matching for semisupervised learning (SSL). The proposed method outperforms the state-of-the-art supervised models in terms of error compensation for Inertial Measurement Unit data from various evaluation metrics and improves the baselines for yaw estimation on average by 38% and 28% across the benchmarks for 30% and 50% of labeled data, respectively. Our approach can be adapted to any SSL approach to compensate for the problem of label scarcity.

Index Terms—Sensor signal processing, contrastive features, noise suppression, reinforcement learning, semisupervised learning.

I. INTRODUCTION

The presence of noise, regardless of the signal type, is ubiquitous in signal processing, therefore leveraging the error compensation methods before an inference is inevitable. Noise suppression methods are divided into conventional approaches and data-driven models. The conventional methods for signal denoising, such as filtering and wavelet transforms, are widely used in industry and academia. Methods such as the Wiener filter [1] work well in removing low-frequency noises like biases, however, they failed in high-frequency noise removal. In the same way, error reduction methods based on a linear model for error compensation like Kalman filter [2] are obviously not applicable for large and complex errors. In addition, the methods such as wavelet processing rely on delicate analysis to choose the basis for signal decomposition, which requires prior knowledge of the nature of signals. Data-driven models recently were leveraged to perform the noise removal task. The deep neural modules have recently seen success to some extent in noise removal for datasets with clean ground truth labels [3], [4]. The existing methods typically employ encoder–decoder architecture for signal denoising and the reconstruction error is used for training and performance evaluation of the deep models. In [5], an unsupervised method based on autoencoders is proposed to remove noise artifacts from IMU signals. The method results in a competent F1-score for human-activity recognition. In [4], an adversarial model is introduced to distinguish the distribution of clean and noisy data toward the target of aligning the distribution of the latent representation of clean and noisy signals. In [6], dilated convolutional layers are applied for denoising gyroscope data, and it was shown that if the model is trained with a proper loss function, resulting accuracy competes with the accuracy of highly accurate methods like visual-inertial systems. Similarly, in [7] a deep model is constructed by using the temporal convolutional network for noise compensation of IMU gyroscope data.

One of the main barriers in data-driven noise suppression approaches is expensive ground-truth noise-free annotations. The existing data-driven approaches mainly require accurate supervision information, such as precise quaternion from highly accurate methods like visual-inertial systems. Regarding training with fewer labels, semisupervised learning (SSL) has been a powerful approach, mitigating the requirement for labeled data. Recent years have witnessed significant progress in semisupervised classification for image domain [8], [9], [10]. However, only a few works address semisupervised noise suppression models with most of them leveraging domain-specific knowledge. The recent SSL approaches split the data augmentation schemes in the image domain into the subcategories of strong and weak augmentations. It is not straightforward to extend these methods to other data modalities. For instance, in sensory multivariate data, the dominant augmentation is adding random noise, which is difficult to divide into strong and weak sets. For instance, adding a random noise with a large mean value for strong augmentation can intensify the noise modalities in the raw sensory data. In this letter, we extend the flexibility of SSL for noise suppression by incorporating reinforcement learning to learn the contrastive features instead of applying strong augmentation of unlabeled samples. In particular, we propose DeDRLSSL based on deep neural models for noise suppression. Our SSL framework is inspired by semisupervised approaches [9], [10], which is composed of supervised loss and consistency loss between the contrastive features of unlabeled samples. We utilize a deep reinforcement learning (DRL) component to refine the augmented data that are used as contrastive features in aligning the model outputs of unlabeled inputs. In particular, we employ DRL to give us a prior of useful augmented contrastive features as a reward to train our framework in SSL fashion. Our approach can be applied to reduce high-frequency errors such as nonlinear random noise and low-frequency errors like biases and scale factors. We empirically demonstrate that the RL component is crucial to achieving improved performances for the noise suppression task.

II. MODEL

In this letter, we address the problem of noise suppression in a semisupervised setting, where we have only a small set of labeled data $\{x^l\}$ with ground truth labels $\{y^l\}$ and a larger set of unlabeled samples $\{x^u\}$. Fig. 1 shows the overview of our architecture. The proposed model takes raw noisy data as the input and regresses the error-compensated data. In the proposed deep model, the error

Corresponding author: Ehsan Kazemi (e-mail: ehsan_kazemy@knights.ucf.edu).
Associate Editor: L. Azpilicueta.
Digital Object Identifier 10.1109/LSENS.2023.3264998

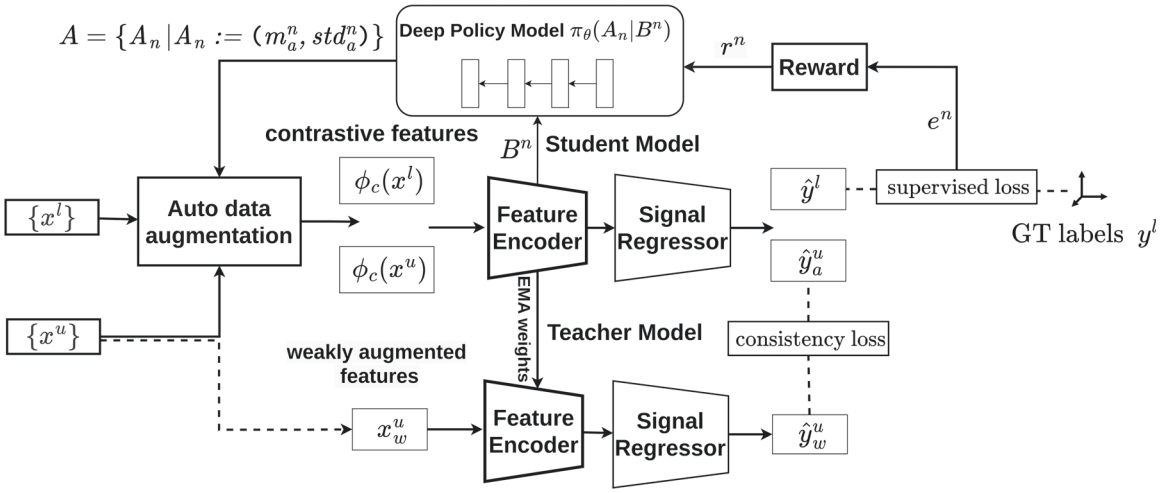


Fig. 1. Overview of training DeDRLSSL model.

features are extracted from a model, and further, the output for error compensation is regressed. In our approach, the cloned teacher model is fed by weakly augmented unlabeled samples while the student model takes augmented unlabeled samples using the parameters from the DRL model. The parameters of the teacher model are the exponential moving average (EMA) of the student model parameters that are learned in the training procedure. We facilitate the correct information flow to the student model by augmenting the input data with random noise with specific standard deviation (std) and noise mean value (m). The selection of the parameters for data augmentation to obtain contrastive features in consistency matching is performed by DRL model, while for weakly augmented examples the parameters mean and variance of random Gaussian noise are fixed. We adopt the SSL model by applying the operator ϕ_c, which sequentially uses the output parameters of the random distribution from the RL model to augment the input data by adding random noise to obtain contrastive features in SSL.

Active noise suppression framework: The data-driven denoisers are typically trained on limited annotated data and fail to generalize to the unseen data that are collected using a different setting. In SSL, data augmentation is used to increase the diversity of the training data that is further used to align the contrastive features for unlabeled data. Our proposed framework utilizes the RL agent to select hyperparameters for noise augmentation for consistency matching in SSL. Later, we will demonstrate that utilizing DRL method to estimate the parameters of the noise distribution for deriving contrastive features in SSL leads to a significant improvement in performance.

Algorithm 1: Training Algorithm for DeDRLSSL.

Input: reward threshold τ, reward penalty parameter ϵ, unsupervised loss weight λ, momentum parameter for EMA γ, number of iterations N, labeled dataset $D_l = \{x^l\}$, unlabeled dataset $D_u = \{x^u\}$
Initialize: θ, f, A_0
Set the EMA model $g = f$
for $n = 0, 1, \ldots, N$ **do**
  for x^l in D_l and x^u in D_u **do**
    Apply A_n to obtain the feature operator $\phi_c(\cdot)$
    Derive the feature map B^n from the feature encoder
    Obtain x_w^u from the weak augmentation of x^u
    Compute the loss L from (2) and update f with $\nabla_f L$
    Calculate the error e^n and determine r^n from (1)
    Update θ with $\mathbb{E}_{\pi_\theta}[r^n \nabla_\theta \log \pi_\theta(A_n|B^n)]$
    Select $A_{n+1} \sim \pi_\theta(\cdot|B^n)$
  end for
  Update g with $\gamma g + (1-\gamma) f$
  Adjust the learning rate based on the scheduler
end for

A. Reinforcement Learning Model

The parameters in the noise distribution of the augmentation operator are initiated randomly. The reward for DRL is computed from the errors of the regressor on annotated data. As described earlier, the agent regresses the parameters for a random noise distribution as an action, which is a pair consisting of the mean and the *std* of random noise. In particular, the contrastive feature $\phi_c(x)$ is obtained from the input x by adding random noise from a random distribution, which is determined by the agent.

State-Action Representation: The state at step n is the triplet (B^n, e^n, A_n), where B^n is the feature map at step n learned by feature encoder, and e^n is the reconstruction error defined by $e^n = \|x^n - x^{gt}\|$, where x^{gt} is the ground truth signal. A_n is the continuous action consisting of the unknown parameters in the random distribution. We use a deep stochastic policy $\pi_\theta(A_n|B^n)$ for action selection, where θ is the parameter that predicts the action $A_n = (m_a^n, std_a^n)$, which indicates the mean m_a^n and std std_a^n of a random distribution to obtain the contrastive feature $\phi_c(\cdot)$. To generate action probabilities, the feature map from the baseline model is fed to the deep neural model, which is shared between the mean selection and std selection model. We let z_a^n denote the output from the penultimate layer of DRL model at time step n. The support of mean and std estimation is tailored to the dataset and the parameters are obtained using the following expression: $m_a(w_a^T z_a^n + b_a) = 10^{\alpha_m \tanh(w_a^T z_a^n + b_a) - \beta_m 1}$, where $[-\beta_m, \alpha_m - \beta_m]$ is the power range for the base 10, specifying the support of mean value distribution. The learnable parameters w_a and b_a are corresponding to the weight and bias of the neural layer. Correspondingly, the distribution for the variance is provided by $std_a(w_a^T z_a^n + b_a) = \exp(\alpha_v \tanh(w_a^T z_a^n + b_a))$. We choose the range of *std* to be more limited than the mean for numerical stability.

Reward Function: We compute the agent rewards from the reconstruction error e^n of the model output at step n. The reward is estimated proportional to the negative values of the reconstruction error while applying an extreme penalty $-\epsilon$ for large ϵ values for the error outliers determined by the threshold τ. More specifically, the reward is given by

$$r_n = \begin{cases} -e^n & \text{if } e^n < \tau \\ -\epsilon & \text{otherwise.} \end{cases} \tag{1}$$

B. Training Protocol

Our training protocol consists of two phases: a supervised training phase, where our denoising regressor model and RL model are trained over labeled data, followed by an SSL stage, which incorporates unlabeled data.

Pretraining stage. We initialize the supervised training procedure on the labeled set with the supervised loss. We clone our model, which is regarded as a teacher model as opposed to the student model. We also train our deep RL model in this stage by utilizing the output actions for a random noise distribution to obtain the contrastive features $\phi_c(x)$ by augmenting the input data.

SSL Training: In this phase, we apply unlabeled data alongside labeled data. The training batch consists of both labeled samples and unlabeled samples and thereby the loss function is composed of the supervised loss for labeled data and the consistency loss for unlabeled data as

$$L = L_l(x^l, y^l) + \lambda L_u(x^u, x^u_w). \tag{2}$$

In this expression, L_l is the supervised loss, L_u is the consistency loss, and λ is the unsupervised loss weight. The loss terms are selected based on domain-specific knowledge.

Algorithm 1 summarizes the training procedure of DeDRLSSL.

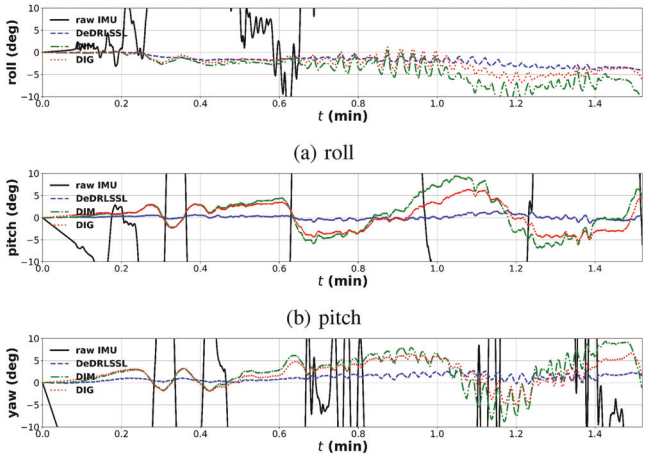
III. EXPERIMENTS

In this letter, we perform experiments on denoising IMU signals to estimate the device orientation. An IMU measures the angular velocity and acceleration of the carrier, which are employed for the carrier dead-reckoning [11]. The measurements from low-cost IMUs suffer from extreme noises such as scale factors, time-dependent offsets, and noise artifacts, especially in free-mode body sensors. The orientation R_n from local frame to global frame at timestamp n is obtained using the single integration of angular velocity ω_n as follows:

$$R_n = R_{n-1} \exp(\omega_n dt) \tag{3}$$

where $\exp(\cdot)$ is SO(3) exponential map and dt is the time increment interval. It is evident from (3) that orientation computation from noisy gyroscope data is prone to error accumulation in time.

Datasets: We evaluate the performance of our proposed method for denoising the IMU data from public datasets EuRoc [12] and TUM-VI [13]. EuRoc consists of micro-aerial vehicle (MAV) data. The IMU data frequency for EuRoc is 200 HZ and the ground truth from the motion capture system is properly time-synchronized with IMU data. The training set is composed of six sequences and four sequences from this dataset are used for testing. TUM-VI consists of inertial measurement sequences from handheld devices, and the ground truth is from the motion capture system. The IMU data are measured at 200 HZ. We use three annotated sequences from this dataset for training and three sequences as a test set.



(a) roll

(b) pitch

(c) yaw

Fig. 2. (a) Roll, (b) pitch, and (c) yaw errors on the EuRoC test sequence V1_03-difficult. The results used 30% of labeled training samples. The solid line is for raw IMU data and the dashed lines are for the denoised data from various approaches.

Baselines: To our best knowledge, we are the first to address the problem of noise suppression in semisupervised procedures in this setting; hence, the comparisons are made with fully supervised baselines. We compare our approach with the state-of-the-art data-driven error compensation approaches DIM [6] and DIG [7] in Table 1. Our goal is to indicate the effectiveness of our framework in SSL setting.

Training setting: Each dataset is divided into training, validation, and test set. We further divide the training split into the labeled set with ground truth and the unlabeled set without ground truth for SSL training, where we set the ratio of labeled data as 50% and 30% of the training set. The denoiser model uses eight dilation convolutional layers and the architecture of RL model consists of two linear layers. All experiments were performed using a single Nvidia V100 GPU on a PC with Core i7 3.60 GHz, 16-GB RAM. We used PyTorch for the implementation of our approach. We use ADAM optimizer with weight decay and the initial learning rate is set to 0.01. We train the baseline models on the labeled split of the training set. We use the validation set to calculate the error e^n. For losses L_l and L_u, we use a loss function based on integrated gyro increments which is recommended in [6]. We also set $\lambda = 0.01$ in the training.

Performance metric: We evaluate our method based on absolute orientation error, where the mean square error between the ground truth and the IMU sequence is computed by

$$Rot_{er} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \left\| \log\left(R_n^T \hat{R}_n\right) \right\|^2}. \tag{4}$$

In this equation, N is the sequence length, $\log(\cdot)$ is the SO(3) logarithm projection, and $\hat{R}_n$ is an estimate of R_n from (3) using $\hat{\omega}_n$, which is the error-compensated gyroscope data at timestamp n. We align estimated orientation and ground truth at $n = 0$. We present Rot_{er} for three quantities roll, pitch, and yaw. We also utilize ROE as [14] with respect to IMU displacement of 8, 16, 24, 34, and 42 m.

Evaluation: Table 1 shows the results of our method compared with baselines under different labeled data ratios. The results show that our proposed method consistently outperforms the other calibration methods in all shown metrics. For example, DeDRLSSL improves the baselines for estimating yaw on average by 38% and 28% correspondingly for 30% and 50% of labeled data. In Fig. 2, we compare the

TABLE 1. Comparison of Baseline and DeDRLSSL Model for Orientation Estimation Error Rot_{er} (Roll, Pitch, Yaw), in Degree

Labeled data percentage		30 %									50 %								
Dataset	Test Subject	DIM Full Sup.			DIG Full Sup.			DeDRLSSL SSL			DIM Full Sup.			DIG Full Sup.			DeDRLSSL SSL		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
EuRoc	MH 02 easy	1.76	3.86	7.42	1.32	6.31	10.36	**0.53**	**1.82**	**2.11**	**0.36**	1.08	1.64	1.35	3.59	5.82	0.90	**0.73**	**1.07**
	MH 04 difficult	2.51	3.40	6.65	2.20	4.22	7.67	**1.16**	**1.39**	**2.31**	1.22	1.27	2.77	1.15	2.60	4.81	**1.04**	**0.91**	**2.04**
	V2 02 medium	6.27	3.39	3.51	4.63	**3.24**	**2.40**	**2.57**	3.58	4.53	5.48	3.81	**2.88**	3.03	3.19	2.95	**2.47**	**3.12**	3.99
	V1 03 difficult	5.35	4.32	4.49	3.20	3.04	3.57	**2.53**	**0.50**	**1.50**	3.81	3.57	3.82	**1.07**	1.62	1.86	2.62	**0.68**	**1.57**
	V1 01 easy	5.75	5.28	8.89	4.62	4.77	8.33	**1.81**	**1.33**	**3.60**	3.70	4.02	6.96	2.30	3.29	5.79	**1.82**	**1.70**	**3.50**
TUM-VI	room 2	1.06	1.06	4.01	1.35	1.58	5.11	**0.71**	**0.73**	**2.08**	0.76	0.78	**1.80**	1.28	1.46	6.84	**0.67**	**0.65**	2.25
	room 4	0.90	0.96	1.68	0.71	**0.64**	1.28	**0.69**	0.80	**1.15**	0.77	0.71	1.09	0.71	0.65	1.01	**0.54**	**0.49**	**0.43**
	room 6	**0.78**	**0.67**	2.92	1.06	1.01	2.82	**0.78**	0.76	**1.89**	**0.79**	0.72	**2.04**	0.99	0.90	2.61	0.82	**0.71**	2.15

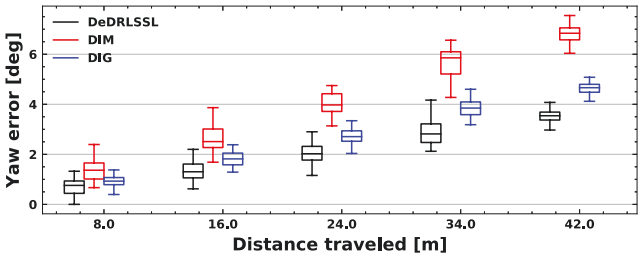


Fig. 3. Relative orientation error (ROE) in terms of yaw errors on the test sequence V1_01-easy from EuRoc. The mean error increases by the distance traveled.

TABLE 2. Comparison of Baselines and DeDRLSSL Models for Orientation Estimation Error Rot_{er} (Roll, Pitch, Yaw)

Dataset	Test Subject	DeDRLSSL w/o DRL model			DeDRLSSL		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw
EuRoc	MH 02 easy	**0.63**	3.86	6.28	0.90	**0.73**	**1.07**
	MH 04 difficult	1.73	2.70	5.08	**1.04**	**0.91**	**2.04**
	V2 02 medium	3.25	**2.99**	**2.89**	**2.47**	3.12	3.99
	V1 03 difficult	3.73	1.84	2.76	**2.62**	**0.68**	**1.57**
	V1 01 easy	3.32	3.31	6.01	**1.82**	**1.70**	**3.50**
TUM-VI	room 2	0.80	0.84	3.23	**0.67**	**0.65**	**2.25**
	room 4	0.72	0.69	1.18	**0.54**	**0.49**	**0.43**
	room 6	**0.78**	0.73	2.25	0.82	**0.71**	**2.15**

roll, pitch, and yaw errors for a test sequence. This figure shows that raw IMU quickly deviates from the ground truth data. In addition, the baseline models deviate from the ground truth in less than 30 s. However, the results from our DeDRLSSL lead to more accurate estimates, being least affected by time. This indicates SSL approach can largely enhance the accuracy of estimations from the baseline models. Fig. 3 shows ROE errors with different IMU displacements, where statistics such as the median and percentiles are calculated. The figure shows that the yaw error obtained by our method is smaller and less affected by the distance traveled, which indicates that the proposed method could effectively compensate for the gyro error.

Ablation Study: Here, we analyze how our DeDLSSE is dependent on the DRL model. Shown in Table 2, the SSL model with DRL model performs noticeably better. Therefore, in this setting applying RL in learning contrastive features leads to additional performance gains compared to utilizing simply noise-augmented data to derive contrastive features.

IV. CONCLUSION

In this letter, we introduce DeDRLSSL: a semisupervised error compensation model. Our model uses deep reinforcement learning for learning contrastive features in semisupervised training and relies on the student–teacher protocol for training. The experimental results from our approach improved the baselines on average by 38% over employing 30% of labeled data. This shows that the existing competing methods rely significantly on accurate supervision information while our SSL approach can compensate for the bottleneck of the amount of labeled data.

REFERENCES

[1] A. Antoniou, *Digital Signal Processing*. Toronto, ON, Canada: McGraw-Hill, 2006.
[2] F. Ghanipoor, M. Hashemi, and H. Salarieh, "Toward calibration of low-precision MEMS IMU using a nonlinear model and TUKF," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4131–4138, Apr. 2020.
[3] D. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Denoising criterion for variational auto-encoding framework," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2059–2065.
[4] L. Casas, A. Klimmek, N. Navab, and V. Belagiannis, "Adversarial signal denoising with encoder-decoder networks," in *Proc. 28th Eur. Signal Process. Conf.*, 2021, pp. 1467–1471.
[5] S. Mohammed and I. Tashev, "Unsupervised deep representation learning to remove motion artifacts in free-mode body sensor networks," in *Proc. IEEE 14th Int. Conf. Wearable Implantable Body Sensor Netw.*, 2017, pp. 183–188.
[6] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising IMU gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4796–4803, Jul. 2020.
[7] F. Huang, Z. Wang, L. Xing, and C. Gao, "A MEMS IMU gyroscope calibration method based on deep learning," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 1003009.
[8] B. Du, T. Xinyao, Z. Wang, L. Zhang, and D. Tao, "Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1440–1453, Apr. 2019.
[9] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 596–608.
[10] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Self-supervised Wasserstein pseudo-labeling for semi-supervised image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12267–12277.
[11] B. Rao, E. Kazemi, Y. Ding, D. M. Shila, F. M. Tucker, and L. Wang, "CTIN: Robust contextual transformer network for inertial navigation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 5, 2022, pp. 5413–5421.
[12] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.
[13] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1680–1687.
[14] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.