
rPPG-Toolbox: Deep Remote PPG Toolbox

Xin Liu¹, Girish Narayanswamy^{1*}, Akshay Paruchuri^{3*}, Xiaoyu Zhang², Jiankai Tang²,
Yuzhe Zhang², Yuntao Wang², Soumyadip Sengupta³, Shwetak Patel¹, Daniel McDuff¹

University of Washington Seattle ¹

Tsinghua University ²

University of North Carolina at Chapel Hill ³

{xliu0, girishvn, dmcudff}@cs.washington.edu, akshay@cs.unc.edu

* Equal Contribution

Abstract

Camera-based physiological measurement is a fast growing field of computer vision. Remote photoplethysmography (rPPG) utilizes imaging devices (e.g., cameras) to measure the peripheral blood volume pulse (BVP) via photoplethysmography, and enables cardiac measurement via webcams and smartphones. However, the task is non-trivial with important pre-processing, modeling, and post-processing steps required to obtain state-of-the-art results. Replication of results and benchmarking of new models is critical for scientific progress; however, as with many other applications of deep learning, reliable codebases are not easy to find or use. We present a comprehensive toolbox, rPPG-Toolbox, that contains unsupervised and supervised rPPG models with support for public benchmark datasets, data augmentation, and systematic evaluation: <https://github.com/ubicomplab/rPPG-Toolbox>

1 Introduction

The vision of ubiquitous computing is to embed computation into everyday objects to enable them to perform useful tasks. The sensing of physiological vital signs is one such task and plays an important role in how health is understood and managed. Cameras are both ubiquitous and versatile sensors, and the transformation of cameras into accurate health sensors has the potential to make the measurement of health signals more comfortable and accessible. Examples of the applications of this technology include systems for monitoring neonates [1], dialysis patients [2], and the detection of arrhythmias [3].

Building on advances in computer vision, camera-based measurement of physiological vital signs has developed into a research field of its own [4]. Researchers have developed methods for measuring cardiac and pulmonary signals by analyzing skin pixel change over time. Recently, several companies have been granted FDA De Novo status for products that use software algorithms to analyze video and estimate pulse rate, heart rate, respiratory rate and/or breathing rate^{1,2}.

There are hundreds of computational architectures that have been proposed for the measurement of cardiopulmonary signals. Unsupervised signal processing methods leverage tools such as Independent Component Analysis (ICA) or Principal Component Analysis (PCA) and assumptions about the periodicity or structure of the underlying blood volume pulse waveform. Neural network architectures can be trained in a supervised fashion using videos with synchronized gold-standard ground truth signals [5–8]. Innovative data generation [9] and augmentation [10], meta-learning for personalization [11, 12], federated learning [13], and unsupervised pretraining [14–17] have been widely explored in the field of camera-based physiological sensing and have led to significant improvements in state-of-the-art performance.

¹https://www.accessdata.fda.gov/cdrh_docs/reviews/DEN200019.pdf

²https://www.accessdata.fda.gov/cdrh_docs/reviews/DEN200038.pdf

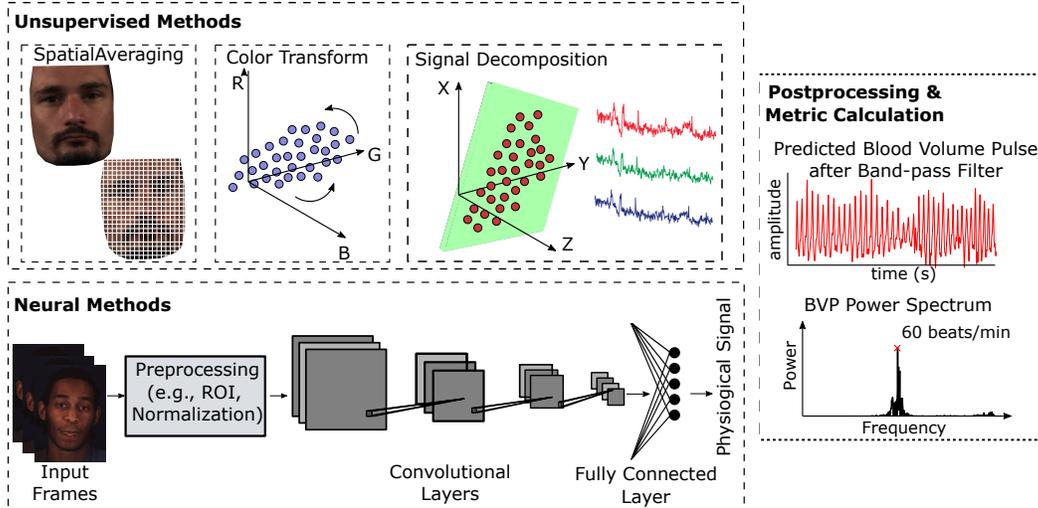


Figure 1: **rPPG Pipeline**. An example of the components of an rPPG pipeline including preprocessing, training, inference, and evaluation.

However, standardization in the field is still severely lacking. Based on our review of literature in the space, we identified four issues that have hindered the interpretation of results in many papers. First, and perhaps most obviously, a lot of the published work is not accompanied by public code. While publishing code repositories with papers is now fairly common in the machine learning and computer vision research communities, it is far less common in the field of camera-based physiological sensing. While there are reasons that it might be difficult to release datasets (e.g., medical data privacy), we cannot find good arguments for not releasing code. Second, many papers do not compare to previously published methods in an “apples-to-apples” fashion. This point is a little more subtle, but rather than performing systematic side-by-side comparisons between methods, the papers compare to numerical results from previous work, even if the training sets and/or test sets are not identical (e.g., test samples were filtered because they were deemed to not have reliable labels). Unfortunately, this often makes it unclear if performance differences are due to data, pre-processing steps, model design, post-processing, training schemes and hardware specifications, or a combination of the aforementioned. Continuing this thread, the third flaw is that papers use pre- and post-processing steps that are not adequately described. Finally, different researchers compute the “labels” (e.g., HR frequency) using their own methods from the contact PPG or ECG timeseries. Differences in these methods lead to different labels and a fundamental issue when it comes to benchmarking performance. When combined, the aforementioned issues make it very difficult to draw conclusions from the literature about the optimal choices for the design of rPPG systems.

Open source code allow researchers to compare novel approaches to consistent baselines without ambiguity regarding the implementation or parameters used. This transparency is important as subsequent research invariably builds on prior state-of-the-art. Implementing a prior method from a paper, even if clearly written, can be difficult. Furthermore, it is an inefficient use of time for many researcher to re-implement all baseline methods. In an effort to address this, several open source toolboxes have been released for camera-based physiological sensing. These toolboxes have been a significant contribution to the community and provide implementations of methods and models [18–20]³ implemented a set of source separation methods (Green, ICA, CHROM, POS) and Pilz [20] published the PPGI-Toolbox⁴ containing implementations of Green, SSR, POS, Local Group Invariance (LGI), Diffusion Process (DP) and Riemannian-PPGI (SPH) models. These toolboxes are implemented in MATLAB (e.g., [18]) and do not contain examples of supervised methods. Python and supervised neural models are now the focus of a large majority of computer vision and deep learning research. There are

³<https://github.com/danmcduff/iphys-toolbox>

⁴<https://github.com/partofthestars/PPGI-Toolbox>

Table 1: **Comparison of rPPG Toolboxes.** Comparison of rPPG-Toolbox with existing toolboxes in camera-based physiological measurement.

Toolbox	Dataset Support	Unsup. Eval	DNN Training	DNN Eval
iPhys-Toolbox [21]	✗	✓	✗	✗
PPG-I Toolbox [20]	✗	✓	✗	✗
pyVHR [19,22]	✓	✓	✗	✓
rPPG-Toolbox (Ours)	✓	✓	✓	✓

Unsup. = Unsupervised learning methods, DNN = Deep neural network methods.

several implementations of popular signal processing methods in Python: Bob.rrpg.base⁵ includes implementations of CHROM, SSR and Boccignone et al. [19] released code for Green, CHROM, ICA, LGI, PBV, PCA, and POS. Several published papers have included links to code; however, often this is only inference code and not training code for neural models. Without providing training code for neural networks, it is challenging for researchers to conduct end-to-end reproducible experiments and build on the research.

In this paper, we present an end-to-end toolbox⁶ for camera-based physiological measurement. This toolbox includes: 1) support for six public datasets, 2) pre-processing code to format the datasets for training neural models, 3) implementations of five neural model architectures and six unsupervised learning methods, 4) evaluation and inference pipelines for supervised and unsupervised learning methods for reproducibility and 5) enabling advanced neural training and inference such as weakly supervised pseudo labels, motion augmentation and multitask learning. We use this toolbox to publish clear and reproducible benchmarks that we hope will provide a foundation for the community to compare methods in a more rigorous and informative manner.

2 Related Work

In the field of remote PPG sensing, there are three significant open-source toolboxes:

iPhys-Toolbox [18]: An open-sourced toolbox written in MATLAB that is comprised of implementations of numerous algorithms for rPPG sensing. It empowers researchers to present results on their datasets using public, standard implementations of baseline methods, ensuring transparency of parameters. This toolbox incorporates a wide range of widely employed baseline methods; however, it falls short on Python support, public dataset data loaders, and neural network training and evaluation.

PPG-I Toolbox [20]: This toolbox provides MATLAB implementations, specifically for six unsupervised signal separation models. It incorporates four evaluation metrics, including Pearson correlation, RMSE/MSE, SNR, and Bland-Altman plots. However, similar to the iPhys-Toolbox, it lacks support for public dataset data loading and neural network training and evaluation.

pyVHR [22]: The most recent in the field, this toolbox adopts Python instead of MATLAB. While it offers ample support for numerous unsupervised methods, its capabilities are limited when it comes to modern neural networks. Notably, pyVHR supports only two neural networks for inference, and none for model training. This omission can be a roadblock for researchers aiming to reproduce and further advance state-of-the-art neural methods.

3 The rPPG-Toolbox

To address the gaps in the current tooling and to promote reproducibility and clearer benchmarking within the camera-based physiological measurement community (rPPG), we present an open-source toolbox designed to support six public datasets, six unsupervised methods and five neural methods for data preprocessing, neural model training and evaluation, and further analysis.

3.1 Datasets

The toolbox includes pre-processing code that converts six public datasets into a form amenable for training with neural models. The standard form for the videos we select includes raw frames and

⁵<https://pypi.org/project/bob.rrpg.base/>

⁶<https://github.com/ubicomplab/rPPG-Toolbox>

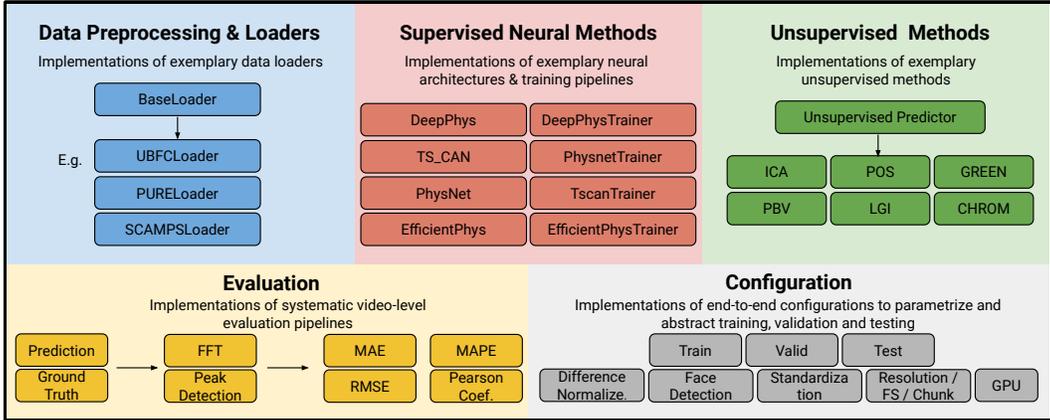


Figure 2: **Overview.** An overview of the rPPG-Toolbox codebase.

difference frames (the difference between each pair of consecutive frames) stored as numpy arrays in a $[N, W, H, C]$ format. Where N is the length of the sequence, W is the width of the frames, H is the height of the frames, and C is the number of channels. There are six channels in this case, as both the raw frames and difference frames account for three color channels each. For faster data loading, all videos in the datasets are typically broken up into several “chunks” of non-overlapping N (e.g., 180) frame sequences. All of these parameters (N, W, H, C) are easy to change and customize. The PPG waveform labels are stored as numpy arrays in a $[N, 1]$ format. The entire pre-processing procedure is supported with multi-thread processing to accelerate the data processing time.

We have provided pre-processing code for UBFC-rPPG [23], PURE [24] SCAMPS [25], MMPD [26], BP4D+ [27], and UBFC-Phys [28]. Each of these datasets encompasses a diverse array of real-world conditions, capturing variations in factors such as motion, lighting, skin tones/types, and backgrounds, thus presenting robust challenges for any signal processing and machine learning algorithms. Tools (Jupyter Notebooks) are provided for quickly visualizing pre-processed datasets and will be detailed further in the provided supplementary materials. We also support the pre-processing and usage of augmented versions of the UBFC-rPPG [23] dataset, which we describe in Section 4.2.

UBFC-rPPG [23]: This dataset features RGB videos recorded using a Logitech C920 HD Pro webcam at 30Hz. The videos have a resolution of 640x480, and they are stored in an uncompressed 8-bit RGB format. Reference PPG data was obtained using a CMS50E transmissive pulse oximeter, thereby providing the gold-standard validation data. The subjects were positioned approximately one meter away from the camera during the recording sessions. The videos were captured under indoor conditions with a combination of natural sunlight and artificial illumination.

PURE [24]: This dataset consists of recordings from 10 subjects, including 8 males and 2 females. The video footage was captured with an RGB eco274CVGE camera from SVS-Vistek GmbH, with a frequency of 30Hz and a resolution of 640x480. Subjects were positioned approximately 1.1 meters from the camera and were illuminated from the front by ambient natural light filtering through a window. The gold-standard ground truth of PPG and SpO2 were obtained at 60Hz with a CMS50E pulse oximeter affixed to the subject’s finger. Each participant completed six recordings under varied motion conditions, thereby contributing to a range of data reflecting different physical states.

SCAMPS [25]: This dataset encompasses 2,800 video clips, comprising 1.68M frames, featuring synthetic avatars in alignment with cardiac and respiratory signals. These waveforms and videos were generated by employing a sophisticated facial processing pipeline, resulting in high-fidelity, quasi-photorealistic renderings. To provide robust test conditions, the videos incorporate various confounders such as head motions, facial expressions, and changes in ambient illumination.

MMPD [26]: This dataset includes 660 one-minute videos recorded using a Samsung Galaxy S22 Ultra mobile phone, at 30 frames per second with a resolution of 1280x720 pixels and then compressed to 320x240 pixels. The ground truth PPG signals were simultaneously captured using an HKG-07C+ oximeter, at 200 Hz and then downsampled to 30 Hz. It contains Fitzpatrick skin types 3-6, four different lighting conditions (LED-low, LED-high, incandescent, natural), four various

activities (stationary, head rotation, talking, and walking), and exercise scenarios. With multiple labels provided, different subsets of this dataset can be easily used for research using our toolbox.

BP4D+ [27]: This dataset contains video footage captured at a rate of 25 frames per second, covering 140 subjects, each participating in 10 emotion-inducing tasks, amounting to a total of 1400 trials. In addition to the standard video footage, the dataset also includes 3D mesh models and thermal video, both captured at the same frame rate. Alongside these, the dataset offers supplementary data including blood pressure measurements (wave, systolic, diastolic, mean), heart rate in beats per minute, respiration (wave, rate bpm), electrodermal activity, and Facial Action Coding System (FACS) encodings for specified action units.

UBFC-Phys [28]: The UBFC-PHYS dataset, a multi-modal dataset, contains 168 RGB videos, with 56 subjects (46 women and 10 men) per a task. There are three tasks with significant amounts of unconstrained motion under static lighting conditions - a rest task, a speech task, and an arithmetic task. The dataset contains gold-standard blood volume pulse (BVP) and electrodermal activity (EDA) measurements that were collected via the Empatica E4 wristband. The videos were recorded at a resolution of 1024x1024 and 35Hz with a EO-23121C RGB digital camera. We utilized all three tasks and the same subject sub-selection list provided by the authors of the dataset in the second supplementary material of Sabour et al. [28] for evaluation. We will reiterate this subject sub-selection list in the supplementary materials of this paper.

3.2 Methods

3.2.1 Unsupervised Methods

The following methods all use linear algebra to recover the estimated PPG signal: 1) **Green** [29]: the green channel information is used as the proxy for the PPG after spatial averaging of RGB video; 2) **ICA** [30]: Independent Component Analysis (ICA) is applied to normalized, spatially averaged color signals to recover demixing matrices; 3) **CHROM** [31]: a linear combination of the chrominance signals obtained from the RGB video are used for estimation; 4) **POS** [32]: plane-orthogonal-to-the-skin (POS), is a method that calculates a projection plane orthogonal to the skin-tone based on physiological and optical principles. A fixed matrix projection is applied to the spatially normalized, averaged pixel values, which are used to recover the PPG waveform; 5) **PBV** [33]: a signature, that is determined by a given light spectrum and changes of the blood volume pulse, is used in order to derive the PPG waveform while offsetting motion and other noise in RGB videos; 6) **LGI** [34]: a feature representation method that is invariant to motion through differentiable local transformations.

3.2.2 Supervised Neural Methods

The following implementations of supervised learning algorithms are included in the toolbox. All implementations were done using PyTorch [37]. Common optimization algorithms, such as Adam [38] and AdamW [39], and criterion, such as mean squared error (MSE) loss, are utilized for training. The learning rate scheduler follows the 1cycle policy [40], which anneals the learning rate from an initial learning rate to some maximum learning rate and then, from that maximum learning rate, to some learning rate much lower than the initial learning rate. The total steps in this policy are determined by the number of epochs specified multiplied by the number of training batches in an epoch. The 1cycle policy allows for convergence due to the learning rate being adjusted well below the initial, maximum learning rate throughout the cycle, and after numerous epochs in which the learning rate is much higher than the final learning rate. We found the 1cycle learning rate scheduler to provide stable results with convergence using a maximum learning rate of 0.009 and 30 epochs. We provide parameters in the toolbox that can enable the visualization of the losses and learning rate changes for both the training and validation phases. Further details on these key visualizations for supervised neural methods will be provided in the supplementary materials.

DeepPhys [5]: A two-branch 2D convolutional attention network architecture. The two representations (appearance and difference frames) are processed by parallel branches with the appearance branch guiding the motion branch via a gated attention mechanism. The target signal is the first differential of the PPG waveform.

PhysNet [6]: A 3D convolutional network architecture. Yu et al. compared this 3D-CNN architecture with a 2D-CNN + RNN architecture, finding that a 3D-CNN version was able to achieve superior pulse rate prediction errors. Therefore, we included the 3D-CNN in this case. It is worth noting that

Table 2: **Benchmark Results.** Performance on the UBFC-rPPG [23], PURE [24] UBFC-Phys [28] and MMPD [26] datasets generated using the rPPG toolbox. For the supervised methods we show cross-dataset training results using the UBFC-rPPG, PURE and SCAMPS datasets.

			Test Set							
			PURE [24]		UBFC-rPPG [23]		UBFC-Phys [28]		MMPD [26]	
	Method	Train Set	MAE \downarrow	MAPE \downarrow						
UNSUPERVISED	GREEN [29]	N/A	10.09	10.28	19.81	18.78	13.55	16.01	21.68	24.39
	ICA [30]	N/A	4.77	4.47	14.70	14.34	10.03	11.85	18.60	20.88
	CHROM [31]	N/A	5.77	11.52	3.98	3.78	4.49	6.00	13.66	15.99
	LGI [34]	N/A	4.61	4.96	15.80	14.70	6.27	7.83	17.08	18.98
	PBV [33]	N/A	3.91	4.82	15.90	15.17	12.34	14.63	17.95	20.18
	POS [32]	N/A	3.67	7.25	4.00	3.86	4.51	6.12	12.36	14.43
SUPERVISED	TS-CAN [7]	UBFC-rPPG	3.69	3.38	N/A	N/A	5.13	6.53	14.00	15.47
		PURE	N/A	N/A	1.29	1.50	5.72	7.34	13.93	15.14
		SCAMPS	4.66	5.83	3.62	3.53	5.55	6.91	19.05	21.77
	PHYSNET [6]	UBFC-rPPG	9.36	17.84	N/A	N/A	5.51	7.47	10.23	12.46
		PURE	N/A	N/A	1.63	1.68	5.07	6.37	13.21	14.73
		SCAMPS	20.08	31.27	4.39	4.39	7.28	9.98	21.05	24.69
		UBFC-rPPG	5.54	5.32	N/A	N/A	6.62	8.21	17.49	19.26
	DEEPHYS [35]	PURE	N/A	N/A	1.21	1.42	8.42	10.18	16.92	18.54
		SCAMPS	3.95	4.25	3.10	3.08	4.75	5.89	15.22	16.56
		UBFC-rPPG	5.47	5.39	N/A	N/A	4.93	6.25	13.78	15.15
	EFF.PHYS-C [36]	PURE	N/A	N/A	2.07	2.10	5.31	6.61	14.03	15.31
		SCAMPS	10.24	11.70	12.64	11.26	6.97	8.47	20.41	23.52

MAE = Mean Absolute Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%).

we used difference-normalized frames as input to PhysNet as the original paper does not specify a concrete input data format.

TS-CAN [7]: A two-branch 2D convolutional attention network architecture that leverages temporal shift operation information across the time axis to perform efficient temporal and spatial modeling. This network is an on-device, real-time algorithm. The target signal is the first differential of the PPG waveform.

EfficientPhys-C [36]: A single-branch 2D convolutional neural network that aims to provide an end-to-end, super lightweight network for real-time on-device computation. The architecture has a normalization module that calculates frame differences and learnable normalization as well as a self-attention module to help the network focus on skin pixels associated with PPG signal.

3.3 Pre-Processing, Training, Post-Processing and Evaluation

In the rPPG-Toolbox, we offer a configuration file system that enables users to modify all parameters used in pre-processing, training, post-processing, and evaluation. A YAML file is provided for every experiment and includes blocks for pre/post-processing, training, validation, testing, model hyperparameters, and computational resources. The pre/post-processing for neural and unsupervised methods share similar settings, such as the same input resolution and face cropping.

In terms of pre-processing, we provide three input data types: 1) "DiffNormalized", which calculates the difference of every two consecutive frames and labels, and normalizes them by their standard deviation; 2) "Standardized", which standardizes the raw frames and labels using z-score; 3) "Raw", which uses the original frames and labels without modification. Additionally, we also provide parameters for face cropping, which is a vital aspect of our task. In the config file, users can use dynamic detection to perform face cropping every N frames and scale the face bounding box by a coefficient to maintain consistency of face cropping in motion videos. Users can also elect to use a median bounding box with dynamic detection in order to help filter out erroneous detections of a face.

With regard to training of neural network, our toolbox provides flexibility to parameterize which portion of the data is used for training / validation / testing. For instance, we can use first 80% of UBFC-rPPG for training, the last 20% of UBFC-rPPG for validation and then use the entire PURE

dataset for testing. Moreover, the distinct parameters (e.g., dropout rate) of each neural network can be defined in the config file.

For post-processing and evaluation, there are several standard post-processing steps that are typically employed to improve model predictions. A 2nd-order Butterworth filter (cut-off frequencies of 0.75 and 2.5 Hz) is applied to filter the predicted PPG waveform. The choice of filtering parameters can have a significant impact on downstream results such as heart rate errors. A Fast Fourier Transform or a peak detection algorithm is then applied to the filtered signal to calculate the heart rate. In this toolbox, we support five metrics for video-level heart rate estimations: mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), signal-noise ratio (SNR), and Pearson Correlation (ρ), along with a calculation of standard error for a better understanding of the accuracy of the aforementioned metrics. We also give users the option to visualize Bland-Altman plots as a part of evaluation. Finer details on the supported metrics, metric results not reported in the main paper, and Bland-Altman plots will appear in the supplementary materials. For better reproducibility, we also provide pre-trained models in our Github repository to allow researchers to perform model inference. The detailed definition of each config parameter is also provided in the Github repository.

3.4 Benchmarking

To show that the implementations of the baseline methods are functioning as expected and provide benchmark results for consumers of the toolbox to reference and reproduce, we performed a set of baseline experiments using three commonly used video rPPG datasets for training: SCAMPS [25], UBFC-rPPG [23] and PURE [24] and tested on four datasets including UBFC-rPPG [23], PURE [24], UBFC-Phys [28], and MMPD [26]. For neural models, a training batch size of 4, 30 epochs, and an inference batch size of 4 was utilized for all experiments. The models were trained using a single Nvidia 2080 Ti GPU. As illustrated in Table 2, we show MAE and MAPE computed between the video-level heart rate estimations and gold standard measurements. Additional metric results can be found in the supplementary materials.

4 Additional Features

4.1 Weakly Supervised Training

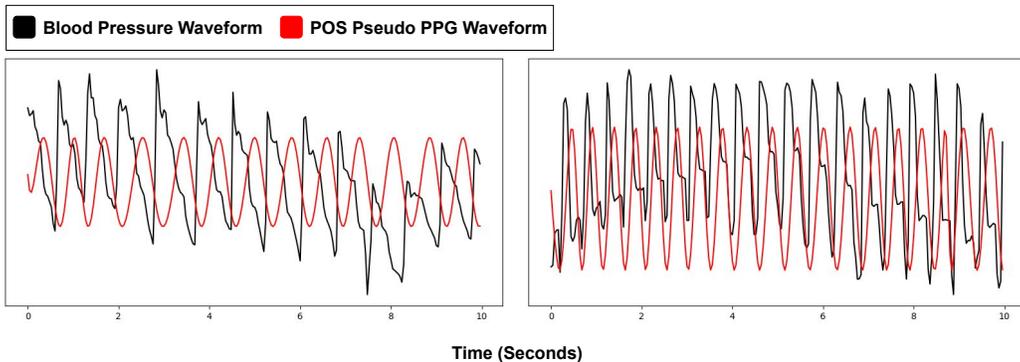


Figure 3: **Generated Pseudo Labels.** Samples of POS [32] generated PPG pseudo labels plotted against ground truth blood pressure waveforms from BP4D+ [27].

Supervised rPPG training requires high fidelity synchronous PPG waveform labels. However not all datasets contain such high quality labels. In these cases we offer the option to train on synchronous PPG "pseudo" labels derived through a signal processing methodology as described by [41]. These labels are produced by using POS-generated [32] PPG waveforms, which are then bandpass filtered around the normal heart-rate frequencies, and finally amplitude normalized using a Hilbert-signal envelope. The tight filtering and envelope normalization results in a strong periodic proxy signal, but at the cost of limited signal morphology.

For instance, in the BP4D+ dataset [27], the cardiac ground truth is represented by a blood pressure waveform. Although this waveform exhibits the same periodicity as the PPG signal, it has a phase shift that adversely affects model training. Figure 3 illustrates sample pseudo labels derived for BP4D+ [27] videos plotted against the ground truth blood pressure waveform. Table 9 presents results for supervised methods, trained on BP4D+ [27] pseudo labels. We extend this feature to all of the supported datasets.

Table 3: **Training with Pseudo Labels.** For the supervised methods we show results training with the (entire) BP4D+ [27] dataset, using POS [32] derived pseudo training labels.

Training Set Testing Set	BP4D+ [27] with POS Pseudo Labels					
	UBFC-rPPG [23]			PURE [24]		
	MAE↓	MAPE↓	ρ ↑	MAE↓	MAPE↓	ρ ↑
Supervised						
TS-CAN [7]	4.69	4.51	0.78	1.29	1.60	0.97
PhysNet(Normalized) [6]	1.78	1.92	0.96	3.69	7.35	0.88
DeepPhys [35]	2.74	2.81	0.93	2.47	2.49	0.89
EfficientPhys-C [36]	2.43	2.52	0.90	3.59	3.27	0.80

MAE = Mean Absolute Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation.

4.2 Motion Augmented Training

The usage of synthetic data in the training of machine learning models for medical applications is becoming a key tool that warrants further research [42]. In addition to providing support for the fully synthetic dataset SCAMPS [25], we provide provide support for synthetic, motion-augmented versions of the UBFC-rPPG [23], PURE [24], SCAMPS [25], and UBFC-PHYS [28] datasets for further exploration toward the use of synthetic data for training rPPG models. The synthetic, motion-augmented datasets are generated using an open-source motion augmentation pipeline targeted for increasing motion diversity in rPPG videos [43]. We present cross-dataset results using a motion-augmented version of the UBFC-rPPG [23] dataset in Table 4. We also provide tools that leverage OpenFace [44] for extracting, visualizing, and analyzing motion in rPPG video datasets. Further details regarding these tools will be shared in both our GitHub repository and our supplementary materials.

Table 4: **Training with Motion-Augmented Data.** We demonstrate results training on a motion-augmented (MA) version of the UBFC-rPPG [23] dataset generated using an open-source motion augmentation pipeline [43] and testing on the unaugmented version of the PURE [24] dataset.

Training Set Testing Set	MAUBFC-rPPG [23]								
	PURE [24]			UBFC-Phys [28]			MMPD [26]		
	MAE↓	MAPE↓	ρ ↑	MAE↓	MAPE↓	ρ ↑	MAE↓	MAPE↓	ρ ↑
TS-CAN [7]	1.07	1.20	0.97	5.03	6.36	0.75	12.59	13.77	0.23
PhysNet (Normalized) [6]	17.03	32.37	0.38	5.51	7.50	0.68	10.67	13.99	0.33
DeepPhys [35]	1.15	1.40	0.97	4.95	6.26	0.75	12.71	13.70	0.21
EfficientPhys-C [36]	2.59	2.67	0.88	4.80	6.10	0.79	13.39	14.50	0.14

MAE = Mean Absolute Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation.

4.3 Extending the rPPG-Toolbox for Physiological Multitasking

While this toolbox is primarily targeted towards rPPG model training and evaluation, it can be easily extended to support multi-tasking of physiological signals. As an example, we implement BigSmall [41], an architecture that multi-tasks PPG, respiration, and facial action. Similar to [41] we present 3-fold cross-validation results across the action unit (AU) subset of BP4D+ [27] (the portion of the dataset with AU labels), and use the same subject-folds and hyper-parameters as implemented in the original publication. These results can be found in Table 5. Note, that like [41], facial action metrics are calculated across 12 common AUs (AU #s 1, 2, 4, 6, 7, 10, 12, 14, 15, 17, 23, 24).

Table 5: **Multitasking Results.** For the BigSmall [41] method we show results for multi-tasking PPG, respiration, and action unit classification; training with the BP4D+ [27] (AU subset) dataset, using POS [32] derived pseudo training PPG labels.

Training Set Testing Set Task	BP4D+ [27]								
	rPPG			Respiration			Facial Action		
	MAE↓	MAPE↓	ρ ↑	MAE↓	MAPE↓	ρ ↑	F1↑	Prec. ↑	Acc. ↑
BigSmall [41]	3.23	3.51	0.83	5.19	26.28	0.14	42.82	39.85	65.73

MAE = Mean Absolute Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation, F1 = average F1 across 12 action units, Prec. = average precision across 12 action units, Acc. = average accuracy across 12 action units.

5 Limitations & Broader Impacts

Although this toolbox supports six datasets and eleven methods, there is still room to expand it to support more recently released datasets and algorithms. Moreover, this toolbox does not support all techniques (e.g., contrastive learning) for pre-training which have been commonly used in many machine learning tasks. Camera sensing has advantages and benefits with the potential to make important cardiac measurement more accessible and comfortable. One of the motivating use-cases for rPPG is turning everyday devices equipped with cameras into scalable health sensors. However, pervasive measurement can also feel intrusive. We are releasing the rPPG toolbox with a Responsible AI License [45] that restricts negative and unintended uses of the toolbox.

6 Conclusion

Research relies on the sharing of ideas. This not only allows methods to be verified, saving time and resources, but also allows researchers to more effectively build upon existing work. Without these resources and open-sourced code bases, fair evaluation and comparison of methods is difficult, creates needless repetitions, and wastes resources. We present an end-to-end and comprehensive toolbox, called rPPG-Toolbox, containing code for pre-processing multiple public datasets, implementations of supervised machine learning (including training pipeline) and unsupervised methods, and post-processing and evaluation tools.

7 Appendix

Our appendices contain the following additional details and results:

- In Section 8 we provide details toward metrics supported by our toolbox. We also provide additional metric results in Section 9 that were not included in the main paper due to space constraints.
- Section 10 briefly details which subjects we utilized for exclusion, or conversely sub-selection, in each task when dealing with the UBFC-Phys [28] dataset. We also briefly describe video filtering criteria available via the toolbox and useful for subject sub-selection.
- Additional details related to training and evaluation for physiological multitasking is shared in Section 11.
- Section 12 briefly describes additional features included in the toolbox . These features, including pre-processed data visualization, loss and learning visualization, Bland-Altman plots, and motion analysis, are further detailed with exemplar usage in the rPPG-Toolbox’s GitHub repo.

8 Metric Details

8.1 rPPG Metrics

We present explanations of metrics supported by our toolbox below.

Mean Absolute Error (MAE): For predicted signal rate R_p , ground truth signal rate R_g , and for N instances:

$$MAE = \frac{1}{N} \sum_{n=1}^N |R_g - R_p|$$

Root Mean Square Error (RMSE): For predicted signal rate R_p , ground truth signal rate R_g , and for N instances:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (R_g - R_p)^2}$$

Mean Absolute Percentage Error (MAPE): For predicted signal rate R_p , ground truth signal rate R_g , and for N instances:

$$MAPE = \frac{1}{N} \sum_{n=1}^N \left| \frac{R_g - R_p}{R_g} \right|$$

Pearson Correlation (ρ): For predicted signal rate R_p , ground truth signal rate R_g , and for N instances, and \bar{R} the average of R for N samples:

$$\rho = \frac{\sum_{n=1}^N (R_{g,n} - \bar{R}_g)(R_{p,n} - \bar{R}_p)}{\sqrt{\left(\sum_{n=1}^N R_{g,n} - \bar{R}_g\right)^2 \left(\sum_{n=1}^N R_{p,n} - \bar{R}_p\right)^2}}$$

Signal-to-Noise Ratio (SNR): As in [31], we calculate the Signal-to-Noise Ratio (SNR) for a predicted signal as the ratio between the area under the curve of the power spectrum around the first and second harmonic of the ground truth heart rate frequency and the area under the curve of the rest of the power spectrum. This is mathematically represented as follows:

$$SNR = 10 \log_{10} \left(\frac{\sum_{45}^{150} (U_t(f)S(f))^2}{\sum_{45}^{150} ((1 - U_t(f))S(f))^2} \right)$$

Where S is the power spectrum of the estimated rPPG signal. $U_t(f)$ is equal to 1 around the first and second harmonics of the ground truth rPPG signal, while being 0 elsewhere in the power spectrum. In the context of the rPPG-Toolbox, only the power spectrum between 0.75 Hz and 2.5 Hz, or 45 beats/min and 150 beats/min, is considered. We report the mean of the SNR values calculated per video or test sample, such that:

$$MSNR = \frac{1}{N} \sum_{n=1}^N SNR$$

Standard Error (\pm SE): The standard error is a measure of the statistical accuracy of an estimate, such as the mean, and is equal to the standard deviation of the theoretical distribution of a large population of such estimates. The standard error takes into account the number of samples utilized in measurement, which is especially useful in the case of remote PPG datasets where the number of test samples can vary significantly from dataset to dataset. For all metrics except for the Pearson correlation (ρ), we calculate the standard error as:

$$SE = \frac{\sigma}{\sqrt{n}}$$

Where σ is the standard deviation and n is the number of samples. For the Pearson correlation (ρ), the standard error is calculated as:

$$SE_{\rho} = \sqrt{\frac{1 - r^2}{n - 2}}$$

Where r is the correlation coefficient and n is the number of samples. Similar to how a standard deviation is reported, we report standard error as $\pm SE$.

8.2 Additional Multitask Metrics

We present explanations of additional metrics added to evaluate the BigSmall [41] model in order to exemplify how this toolbox can be extended to support physiological multitasking.

Evaluated Action Units (AU): Similar to [41], and other AU literature, facial action metrics are calculated for the following 12 commonly used AUs: AU01, AU02, AU04, AU06, AU07, AU10, AU12, AU14, AU15, AU17, AU23, AU24.

F1: The harmonic mean of recall and precision. For true positive count TP , false positive count FP , and false negative count FN .

$$F1 = 100 * \frac{2TP}{2TP + FP + FN}$$

Precision (Prec.): For true positive count TP , and false positive count FP :

$$Precision = 100 * \frac{TP}{TP + FP}$$

Accuracy (Acc.) For true positive count TP , true negative count TN , false positive count FP , and false negative count FN :

$$Accuracy = 100 * \frac{TP + TN}{TP + TN + FP + FN}$$

9 Additional Results

We reiterate results provided in the main paper and present additional results including the RMSE, SNR, Pearson correlation, and the corresponding standard errors. Note that there may be minor differences between results in the following tables and the main paper, as they were generated on a different machine using the latest version of the rPPG-Toolbox.

Table 6: **Benchmark Results.** Performance on the UBFC-rPPG [23], PURE [24] UBFC-Phys [28] and MMPD [26] datasets generated using the rPPG toolbox. For the supervised methods we show cross-dataset training results using the UBFC-rPPG, PURE and SCAMPS datasets.

			Test Set					
			PURE [24]					
	Method	Train Set	MAE↓	RMSE ↓	MAPE↓	ρ ↑	SNR ↑	
UNSUPERVISED	GREEN [29]	N/A	10.09 ± 2.81	23.85 ± 217.81	10.28 ± 2.33	0.34 ± 0.12	-2.66 ± 1.43	
	ICA [30]	N/A	4.77 ± 2.08	16.07 ± 153.84	4.47 ± 1.65	0.72 ± 0.09	5.24 ± 1.77	
	CHROM [31]	N/A	5.77 ± 1.79	14.93 ± 81.53	11.52 ± 3.75	0.81 ± 0.08	4.58 ± 0.85	
	LGI [34]	N/A	4.61 ± 1.91	15.38 ± 134.14	4.96 ± 1.72	0.77 ± 0.08	4.50 ± 1.21	
	PBV [33]	N/A	3.92 ± 1.61	12.99 ± 123.60	4.84 ± 1.49	0.84 ± 0.07	2.30 ± 1.31	
	POS [32]	N/A	3.67 ± 1.46	11.82 ± 66.87	7.25 ± 3.03	0.88 ± 0.06	6.87 ± 0.95	
SUPERVISED	TS-CAN [7]	UBFC-rPPG	3.69 ± 1.74	13.8 ± 113.84	3.39 ± 1.44	0.82 ± 0.08	5.26 ± 1.11	
		SCAMPS	4.66 ± 1.68	13.69 ± 92.53	5.83 ± 2.03	0.82 ± 0.08	0.95 ± 1.04	
	PHYSNET [6]	UBFC-rPPG	9.36 ± 2.39	20.63 ± 116.59	17.84 ± 4.68	0.62 ± 0.10	7.84 ± 1.00	
		SCAMPS	20.08 ± 2.46	27.56 ± 148.91	31.28 ± 3.89	0.09 ± 0.13	-10.21 ± 0.56	
	DEEPPHYS [35]	UBFC-rPPG	5.54 ± 2.30	18.51 ± 173.09	5.32 ± 1.90	0.66 ± 0.10	4.40 ± 1.32	
		SCAMPS	3.96 ± 1.67	13.44 ± 98.86	4.25 ± 1.60	0.83 ± 0.07	5.07 ± 1.15	
	EFF.PHYS-C [36]	UBFC-rPPG	5.47 ± 2.10	17.04 ± 143.80	5.40 ± 1.76	0.71 ± 0.09	4.09 ± 1.16	
		SCAMPS	10.24 ± 2.48	21.65 ± 173.96	11.70 ± 2.28	0.46 ± 0.12	-5.49 ± 1.05	
				Test Set				
				UBFC-rPPG [23]				
	Method	Train Set	MAE↓	RMSE ↓	MAPE↓	ρ ↑	SNR ↑	
UNSUPERVISED	GREEN [29]	N/A	19.73 ± 3.75	31.00 ± 235.38	18.72 ± 3.33	0.37 ± 0.15	-11.18 ± 1.63	
	ICA [30]	N/A	16.00 ± 3.09	25.65 ± 163.58	15.35 ± 2.77	0.44 ± 0.14	-9.91 ± 1.78	
	CHROM [31]	N/A	4.06 ± 1.21	8.83 ± 33.93	3.84 ± 1.10	0.89 ± 0.07	-2.96 ± 1.18	
	LGI [34]	N/A	15.80 ± 3.67	28.55 ± 236.17	14.70 ± 3.20	0.36 ± 0.15	-8.15 ± 1.41	
	PBV [33]	N/A	15.90 ± 3.25	26.40 ± 199.71	15.17 ± 2.91	0.48 ± 0.14	-9.16 ± 1.35	
	POS [32]	N/A	4.08 ± 1.01	7.72 ± 21.87	3.93 ± 0.91	0.92 ± 0.06	-2.39 ± 1.14	
SUPERVISED	TS-CAN [7]	PURE	1.30 ± 0.40	2.87 ± 3.05	1.50 ± 0.47	0.99 ± 0.02	1.49 ± 1.13	
		SCAMPS	3.62 ± 0.91	6.92 ± 18.30	3.53 ± 0.84	0.93 ± 0.06	-3.91 ± 0.98	
	PHYSNET [6]	PURE	1.63 ± 0.53	3.79 ± 7.59	1.68 ± 0.56	0.98 ± 0.03	-1.79 ± 1.11	
		SCAMPS	4.39 ± 1.27	9.31 ± 39.46	4.39 ± 1.21	0.86 ± 0.08	-5.46 ± 0.98	
	DEEPPHYS [35]	PURE	1.21 ± 0.41	2.90 ± 3.75	1.42 ± 0.49	0.99 ± 0.02	1.74 ± 1.16	
		SCAMPS	3.10 ± 1.44	9.81 ± 74.70	3.08 ± 1.32	0.87 ± 0.08	-0.79 ± 1.22	
	EFF.PHYS-C [36]	PURE	2.07 ± 0.92	6.32 ± 32.01	2.10 ± 0.87	0.94 ± 0.05	-0.12 ± 1.20	
		SCAMPS	12.64 ± 3.15	23.99 ± 182.44	11.26 ± 2.67	0.34 ± 0.15	-9.36 ± 1.05	

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation, SNR = Signal-to-Noise Ratio (dB) when comparing predicted spectrum to ground truth spectrum.

Table 7: **Benchmark Results.** Performance on the UBFC-rPPG [23], PURE [24] UBFC-Phys [28] and MMPD [26] datasets generated using the rPPG toolbox. For the supervised methods we show cross-dataset training results using the UBFC-rPPG, PURE and SCAMPS datasets.

			Test Set				
			UBFC-Phys [28]				
	Method	Train Set	MAE↓	RMSE ↓	MAPE↓	ρ ↑	SNR ↑
UNSUPERVISED	GREEN [29]	N/A	13.55 ± 1.30	18.80 ± 48.87	16.01 ± 1.42	0.29 ± 0.10	-10.34 ± 0.65
	ICA [30]	N/A	10.04 ± 1.20	15.73 ± 43.63	11.85 ± 1.35	0.36 ± 0.09	-5.28 ± 0.98
	CHROM [31]	N/A	4.49 ± 0.60	7.56 ± 13.84	6.00 ± 0.88	0.80 ± 0.06	-1.92 ± 0.85
	LGI [34]	N/A	6.27 ± 0.83	10.41 ± 22.76	7.83 ± 0.99	0.70 ± 0.07	-3.30 ± 0.91
	PBV [33]	N/A	12.34 ± 1.22	17.43 ± 47.24	14.63 ± 1.33	0.33 ± 0.09	-9.33 ± 0.71
	POS [32]	N/A	4.51 ± 0.68	8.16 ± 17.36	6.12 ± 0.99	0.77 ± 0.06	-1.28 ± 0.90
SUPERVISED	TS-CAN [7]	UBFC-rPPG	5.13 ± 0.63	8.12 ± 18.47	6.53 ± 0.85	0.76 ± 0.07	-1.95 ± 0.81
		PURE	5.72 ± 0.66	8.78 ± 16.94	7.34 ± 0.90	0.72 ± 0.07	-3.72 ± 0.78
		SCAMPS	5.55 ± 0.67	8.71 ± 16.96	6.91 ± 0.85	0.72 ± 0.07	-4.40 ± 0.66
	PHYSNET [6]	UBFC-rPPG	5.51 ± 0.85	10.18 ± 29.95	7.47 ± 1.26	0.70 ± 0.07	-0.67 ± 1.00
		PURE	5.07 ± 0.90	10.34 ± 31.74	6.37 ± 1.16	0.62 ± 0.08	-1.05 ± 0.95
		SCAMPS	7.13 ± 0.95	11.95 ± 41.44	9.79 ± 1.50	0.51 ± 0.09	-7.53 ± 0.53
	DEEPHYS [35]	UBFC-rPPG	6.62 ± 0.84	10.69 ± 25.90	8.21 ± 1.04	0.66 ± 0.08	-2.98 ± 0.82
		PURE	8.42 ± 1.09	13.80 ± 38.06	10.18 ± 1.29	0.44 ± 0.09	-4.41 ± 0.84
		SCAMPS	4.75 ± 0.58	7.50 ± 14.47	5.89 ± 0.72	0.82 ± 0.06	-2.04 ± 0.76
	EFF.PHYS-C [36]	UBFC-rPPG	4.93 ± 0.58	7.65 ± 14.44	6.25 ± 0.79	0.79 ± 0.06	-2.09 ± 0.82
		PURE	5.31 ± 0.78	9.44 ± 27.67	6.61 ± 0.96	0.70 ± 0.07	-2.22 ± 0.81
		SCAMPS	6.97 ± 0.79	10.58 ± 22.70	8.47 ± 0.91	0.64 ± 0.08	-7.38 ± 0.47
			Test Set				
			MMPD [26]				
	Method	Train Set	MAE↓	RMSE ↓	MAPE↓	ρ ↑	SNR ↑
UNSUPERVISED	GREEN [29]	N/A	21.68 ± 0.67	27.69 ± 42.21	24.39 ± 0.64	-0.01 ± 0.04	-14.34 ± 0.26
	ICA [30]	N/A	18.60 ± 0.61	24.30 ± 33.80	20.88 ± 0.58	0.01 ± 0.04	-13.84 ± 0.27
	CHROM [31]	N/A	13.66 ± 0.50	18.76 ± 23.82	16.00 ± 0.57	0.08 ± 0.04	-11.74 ± 0.21
	LGI [34]	N/A	17.08 ± 0.62	23.32 ± 34.46	18.98 ± 0.60	0.04 ± 0.04	-13.15 ± 0.25
	PBV [33]	N/A	17.95 ± 0.60	23.58 ± 32.45	20.18 ± 0.58	0.09 ± 0.04	-13.88 ± 0.24
	POS [32]	N/A	12.36 ± 0.49	17.71 ± 23.65	14.43 ± 0.55	0.18 ± 0.04	-11.53 ± 0.22
SUPERVISED	TS-CAN [7]	UBFC-rPPG	14.01 ± 0.61	21.04 ± 30.02	15.48 ± 0.61	0.24 ± 0.04	-10.18 ± 0.28
		PURE	13.94 ± 0.64	21.61 ± 33.02	15.15 ± 0.63	0.20 ± 0.04	-9.94 ± 0.27
		SCAMPS	19.05 ± 0.58	24.20 ± 31.90	21.77 ± 0.60	0.14 ± 0.04	-13.24 ± 0.25
	PHYSNET [6]	UBFC-rPPG	10.24 ± 0.51	16.54 ± 23.04	12.46 ± 0.65	0.29 ± 0.04	-8.95 ± 0.24
		PURE	13.22 ± 0.56	19.61 ± 26.13	14.74 ± 0.58	0.23 ± 0.04	-10.94 ± 0.24
		SCAMPS	21.06 ± 0.55	25.38 ± 32.56	24.70 ± 0.62	0.14 ± 0.04	-16.45 ± 0.22
	DEEPHYS [35]	UBFC-rPPG	17.50 ± 0.70	25.00 ± 38.62	19.27 ± 0.68	0.06 ± 0.04	-11.72 ± 0.33
		PURE	16.92 ± 0.70	24.61 ± 38.03	18.54 ± 0.68	0.05 ± 0.04	-11.53 ± 0.31
		SCAMPS	15.22 ± 0.68	23.17 ± 38.46	16.56 ± 0.66	0.09 ± 0.04	-10.23 ± 0.31
	EFF.PHYS-C [36]	UBFC-rPPG	13.78 ± 0.68	22.25 ± 37.94	15.15 ± 0.70	0.09 ± 0.04	-9.13 ± 0.31
		PURE	14.03 ± 0.64	21.62 ± 32.95	15.32 ± 0.63	0.17 ± 0.04	-9.95 ± 0.29
		SCAMPS	20.41 ± 0.57	25.06 ± 31.72	23.52 ± 0.61	0.11 ± 0.04	-14.28 ± 0.24

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation, SNR = Signal-to-Noise Ratio (dB) when comparing predicted spectrum to ground truth spectrum.

Table 8: **Training with Motion-Augmented Data.** We demonstrate results training on a motion-augmented (MA) version of the UBFC-rPPG [23] dataset generated using an open-source motion augmentation pipeline [43] and testing on the unaugmented versions of the PURE [24] dataset, UBFC-Phys [28], and MMPD [26] datasets.

Training Set Testing Set Metric (\pm Std. Err.)	MAUBFC-rPPG [23]				
	PURE [24]				
	MAE \downarrow	RMSE \downarrow	MAPE \downarrow	ρ \uparrow	SNR \uparrow
Supervised					
TS-CAN [7]	1.07 \pm 0.75	5.89 \pm 33.75	1.20 \pm 0.83	0.97 \pm 0.03	8.86 \pm 0.95
PhysNet (Normalized) [6]	17.03 \pm 2.97	28.50 \pm 149.16	32.37 \pm 5.82	0.38 \pm 0.12	7.27 \pm 0.88
DeepPhys [35]	1.15 \pm 0.76	5.95 \pm 33.75	1.40 \pm 0.85	0.97 \pm 0.03	9.94 \pm 1.00
EfficientPhys-C [36]	2.59 \pm 1.43	11.29 \pm 96.01	2.67 \pm 1.27	0.88 \pm 0.06	6.75 \pm 1.12

Training Set Testing Set Metric (\pm Std. Err.)	MAUBFC-rPPG [23]				
	UBFC-Phys [28]				
	MAE \downarrow	RMSE \downarrow	MAPE \downarrow	ρ \uparrow	SNR \uparrow
Supervised					
TS-CAN [7]	5.03 \pm 0.67	8.39 \pm 18.26	6.36 \pm 0.90	0.75 \pm 0.07	-1.15 \pm 0.81
PhysNet (Normalized) [6]	5.51 \pm 0.88	0.44 \pm 37.65	7.50 \pm 1.32	0.68 \pm 0.07	-0.57 \pm 1.08
DeepPhys [35]	4.95 \pm 0.67	8.37 \pm 21.53	6.26 \pm 0.90	0.75 \pm 0.07	-0.78 \pm 0.85
EfficientPhys-C [36]	4.80 \pm 0.58	7.52 \pm 15.02	6.10 \pm 0.79	0.79 \pm 0.06	-0.87 \pm 0.86

Training Set Testing Set Metric (\pm Std. Err.)	MAUBFC-rPPG [23]				
	MMPD [26]				
	MAE \downarrow	RMSE \downarrow	MAPE \downarrow	ρ \uparrow	SNR \uparrow
Supervised					
TS-CAN [7]	12.59 \pm 0.62	20.23 \pm 31.27	13.77 \pm 0.62	0.23 \pm 0.04	-9.19 \pm 0.29
PhysNet (Normalized) [6]	10.68 \pm 0.49	16.56 \pm 19.72	14.01 \pm 0.72	0.32 \pm 0.04	-9.28 \pm 0.21
DeepPhys [35]	12.71 \pm 0.65	21.04 \pm 35.40	13.70 \pm 0.64	0.21 \pm 0.04	-8.85 \pm 0.31
EfficientPhys-C [36]	13.42 \pm 0.66	21.64 \pm 35.46	14.52 \pm 0.65	0.14 \pm 0.04	-9.20 \pm 0.31

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation, SNR = Signal-to-Noise Ratio (dB) when comparing predicted spectrum to ground truth spectrum.

Table 9: **Training with Pseudo Labels.** For the supervised methods we show results training with the (entire) BP4D+ [27] dataset, using POS [32] derived pseudo training labels.

Training Set Testing Set Metric (\pm Std. Err.)	BP4D+ [27] with POS Pseudo Labels				
	UBFC-rPPG [23]				
	MAE \downarrow	RMSE \downarrow	MAPE \downarrow	ρ \uparrow	SNR \uparrow
Supervised					
TS-CAN [7]	4.69 \pm 1.88	13.04 \pm 100.15	4.51 \pm 1.65	0.78 \pm 0.10	0.01 \pm 1.27
PhysNet(Normalized) [6]	1.78 \pm 0.67	4.68 \pm 11.94	1.92 \pm 0.72	0.96 \pm 0.04	1.24 \pm 1.08
DeepPhys [35]	2.74 \pm 0.96	6.78 \pm 27.43	2.81 \pm 0.91	0.93 \pm 0.06	-0.22 \pm 1.33
EfficientPhys-C [36]	2.43 \pm 1.29	8.68 \pm 67.51	2.52 \pm 1.20	0.90 \pm 0.07	0.39 \pm 1.27

Training Set Testing Set Metric (\pm Std. Err.)	BP4D+ [27] with POS Pseudo Labels				
	PURE [24]				
	MAE \downarrow	RMSE \downarrow	MAPE \downarrow	ρ \uparrow	SNR \uparrow
Supervised					
TS-CAN [7]	1.29 \pm 0.76	6.00 \pm 33.74	1.60 \pm 0.86	0.97 \pm 0.03	8.61 \pm 1.02
PhysNet(Normalized) [6]	3.69 \pm 1.46	11.79 \pm 64.42	7.35 \pm 3.01	0.88 \pm 0.06	8.33 \pm 0.06
DeepPhys [35]	2.47 \pm 1.41	11.11 \pm 93.02	2.49 \pm 1.21	0.89 \pm 0.061	7.32 \pm 1.09
EfficientPhys-C [36]	3.59 \pm 1.84	14.55 \pm 135.51	3.27 \pm 1.50	0.80 \pm 0.08	7.48 \pm 1.15

MAE = Mean Absolute Error in HR estimation (Beats/Min), RMSE = Root Mean Square Error in HR estimation (Beats/Min), MAPE = Mean Percentage Error (%), ρ = Pearson Correlation in HR estimation, SNR = Signal-to-Noise Ratio (dB) when comparing predicted spectrum to ground truth spectrum.

Table 10: **Full 3-Fold Multitasking Results.** For the BigSmall [41] method we show the full 3-fold results for multi-tasking PPG, respiration, and action unit classification; training and evaluating on the BP4D+ [27] (AU subset) dataset, using POS [32] derived pseudo training PPG labels.

	Training Set Testing Set Fold	BP4D+ [27]					
		Fold 1	Fold 2	Fold 3			
Heart (Metric \pm Std. Err. \downarrow)	MAE \downarrow	4.24 \pm 0.73		2.91 \pm 0.49		2.54 \pm 0.48	
	RMSE \downarrow	10.76 \pm 33.20		7.26 \pm 13.90		7.06 \pm 16.67	
	MAPE \downarrow	4.55 \pm 0.74		3.22 \pm 0.53		2.75 \pm 0.49	
	$\rho\uparrow$	0.68 \pm 0.05		0.90 \pm 0.03		0.91 \pm 0.03	
	SNR \uparrow	3.85 \pm 0.69		6.27 \pm 0.67		6.53 \pm 0.63	
Respiration (Metric \pm Std. Err. \downarrow)	MAE \downarrow	5.28 \pm 0.31		4.96 \pm 0.33		5.34 \pm 0.35	
	RMSE \downarrow	6.74 \pm 4.38		6.67 \pm 4.96		7.18 \pm 5.12	
	MAPE \downarrow	24.41 \pm 1.55		25.30 \pm 2.08		29.14 \pm 2.72	
	$\rho\uparrow$	0.15 \pm 0.07		0.16 \pm 0.72		0.12 \pm 0.07	
	SNR \uparrow	7.69 \pm 0.64		10.53 \pm 0.75		9.34 \pm 0.64	
Facial Action (AU) (F1 \uparrow , Prec. \uparrow)	AU01	18.62	11.04	18.88	11.34	24.32	16.43
	AU02	20.76	12.73	18.28	10.89	15.46	9.07
	AU04	12.57	8.08	11.48	7.85	14.43	8.63
	AU06	66.73	66.58	64.71	61.09	76.44	79.20
	AU07	74.86	78.68	70.08	75.10	75.58	86.34
	AU10	74.92	77.32	70.09	74.48	82.09	90.34
	AU12	72.69	70.79	67.75	68.54	80.96	88.02
	AU14	67.21	72.84	70.18	69.11	66.73	70.93
	AU15	22.56	13.91	22.33	13.38	29.64	22.13
	AU17	25.77	18.01	20.95	12.45	38.17	28.06
	AU23	34.64	27.41	34.21	24.19	40.68	28.76
AU24	7.00	3.71	10.70	6.20	19.03	10.81	
Facial Action (AU) (Metric Mean)	F1 \uparrow	41.53		39.97		46.96	
	Prec. \uparrow	36.42		36.22		44.89	
	Acc. (%) \uparrow	61.91		62.42		72.83	

For HR estimation, MAE = Mean Absolute Error, RMSE = Root Mean Square Error, MAPE = Mean Percentage Error (%), ρ = Pearson Correlation, SNR = Signal-to-Noise Ratio (dB) when comparing predicted spectrum to ground truth spectrum. For AU classification F1 = harmonic mean of precision and recall, Prec. = precision, Acc. = accuracy.

10 UBFC-Phys Video Exclusion

For evaluation of the UBFC-Phys [28] dataset in our main paper and by default in our toolbox, we utilized all three tasks and the same subject exclusion, or conversely sub-selection, list provided by the authors of the dataset in the second supplementary material of their paper [28]. Based on the aforementioned supplementary material, we eliminated 14 subjects (s3, s8, s9, s26, s28, s30, s31, s32, s33, s40, s52, s53, s54, s56) for the rest task (T1), 30 subjects (s1, s4, s6, s8, s9, s11, s12, s13, s14, s19, s21, s22, s25, s26, s27, s28, s31, s32, s33, s35, s38, s39, s41, s42, s45, s47, s48, s52, s53, s55) for the speech task (T2), and 23 subjects (s5, s8, s9, s10, s13, s14, s17, s22, s25, s26, s28, s30, s32, s33, s35, s37, s40, s47, s48, s49, s50, s52, s53) for the arithmetic task (T3).

In our toolbox, video exclusion is achieved using dataset filtering criteria specified in the config file. Specifically, an exclusion list or a task selection list can be provided to respectively exclude videos from being included or to select specific tasks as a part of a dataset.

11 Multitasking Training and Evaluation Details

To show how this toolbox may be extended for physiological multitasking, we implement BigSmall [41] a model that multitasks PPG, respiration, and facial action. Here we reiterate information from [41], with slight modifications, for clarification.

11.1 Cross Validation Subject Folds

Fold 1: F003, F004, F005, F006, F009, F017, F022, F028, F029, F031, F032, F033, F038, F044, F047, F048, F052, F053, F055, F061, F063, F067, F068, F074, F075, F076, F081, M003, M005, M006, M009, M012, M019, M025, M026, M028, M031, M036, M037, M040, M046, M047, M049, M051, M054, M056.

Fold 2: F001, F002, F008, F018, F021, F025, F026, F035, F036, F037, F039, F040, F041, F042, F046, F049, F057, F058, F060, F062, F064, F066, F070, F071, F072, F073, F077, F082, M001, M002, M007, M013, M014, M016, M022, M023, M024, M027, M029, M030, M034, M035, M041, M042, M043, M048, M055.

Fold 3: F007, F010, F011, F012, F013, F014, F015, F016, F019, F020, F023, F024, F027, F030, F034, F043, F045, F050, F051, F054, F056, F059, F065, F069, F078, F079, F080, M004, M008, M010, M011, M015, M017, M018, M020, M021, M032, M033, M038, M039, M044, M045, M050, M052, M053, M057, M058.

11.2 AU Subset

The AU subset used for model training and evaluation (in this toolbox) is made up of dataset subset which contains action unit labels. This consists of approximately 20 seconds worth of data from the following tasks for each subject: T1, T6, T7, T8.

11.3 Subject Fold Splits

[41] is evaluated using 3 fold cross validation, where the folds are comprised of trials from mutually exclusive subjects in the dataset. These subject-wise folds are outlined below.

12 Additional Features

12.1 Pre-processed Data Visualization

Pre-processing is an important aspect of the rPPG task that we hope to help standardize using our toolbox. It is advantageous to be able to quickly visualize and visually evaluate pre-processed image data and ground truth signals. Image data in particular can be especially useful to observe in order to inspect the effectiveness of out-of-the-box face detection and cropping techniques used in our toolbox, and to ultimately get an idea as to how much of the face region is visible in a given video. We provide simple Jupyter Notebooks for quickly visualizing image data and ground truth signals pre-processed by our toolbox. Further details regarding these notebooks can be found in our GitHub repo and the associated README.

12.2 Training Loss, Validation Loss, and Learning Rate Visualization

The rPPG-Toolbox assumes certain defaults across most config files for supervised methods, including a default learning rate of 0.009 used alongside the Adam [38] or AdamW [39] optimizers, a criterion such as mean squared error (MSE) loss or Negative Pearson Correlation Loss, and the 1cycle learning rate scheduler [40] are utilized for training. An exception is with BigSmall [41], which uses a default learning rate of 0.001 that remains constant throughout training. It can be valuable to visualize losses such as those involved in training or validation phases. Furthermore, it may be useful to simultaneously visualize the learning rate, especially when users stray from the defaults in order to target an optimal set of training, validation, and testing parameters for their research efforts. The toolbox’s configs contain parameters that enable the visualization of the training loss, validation loss, and the learning rate for any given supervised method.

12.3 Bland-Altman Plots

We provide Bland-Altman plots as an additional metric in the rPPG-Toolbox. Users can enable the plots via an evaluation parameter in the config file, and will be given further options to configure the plots as the toolbox is refined and expanded. For more details, please refer to the GitHub repo and the associated README.

12.4 Motion Analysis

We also provide scripts that leverage OpenFace [44] for extracting, visualizing, and analyzing motion in rPPG video datasets. Specifically, we include a Python script to convert datasets into the .mp4 format for subsequent analysis by OpenFace, a shell script that leverages OpenFace to perform both rigid and non-rigid head motion analysis, and a separate Python script that plots exemplar plots that showcase comparisons of motion between different datasets. Further details can be found in our GitHub repo and the associated README.

References

- [1] Lonke AM Aarts, Vincent Jeanne, John P Cleary, C Lieber, J Stuart Nelson, Sidarto Bambang Oetomo, and Wim Verkruyse. Non-contact heart rate monitoring utilizing camera photoplethysmography in the neonatal intensive care unit—a pilot study. *Early human development*, 89(12):943–948, 2013.
- [2] Lionel Tarassenko, Mauricio Villarroel, Alessandro Guazzi, Joao Jorge, DA Clifton, and Chris Pugh. Non-contact video-based vital sign monitoring using ambient light and auto-regressive models. *Physiological measurement*, 35(5):807, 2014.
- [3] Bryan P Yan, William HS Lai, Christy KY Chan, Alex CK Au, Ben Freedman, Yukkee C Poh, and Ming-Zher Poh. High-throughput, contact-free detection of atrial fibrillation from video with deep learning. *JAMA cardiology*, 5(1):105–107, 2020.
- [4] Daniel McDuff. Camera measurement of physiological vital signs. *arXiv preprint arXiv:2111.11547*, 2021.
- [5] Weixuan Chen and Daniel McDuff. DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks. *arXiv:1805.07888 [cs]*, August 2018. arXiv: 1805.07888.
- [6] Zitong Yu, Xiaobai Li, and Guoying Zhao. Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks. *arXiv preprint arXiv:1905.02419*, 2019.
- [7] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *arXiv preprint arXiv:2006.03790*, 2020.
- [8] Zitong Yu, Xiaobai Li, Pichao Wang, and Guoying Zhao. Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection. *IEEE Signal Processing Letters*, 2021.
- [9] Daniel McDuff, Javier Hernandez, Erroll Wood, Xin Liu, and Tadas Baltrusaitis. Advancing non-contact vital sign measurement using synthetic avatars. *arXiv preprint arXiv:2010.12949*, 2020.

- [10] Zhen Wang, Yunhao Ba, Pradyumna Chari, Oyku Deniz Bozkurt, Gianna Brown, Parth Patwa, Niranjana Vaddi, Laleh Jalilian, and Achuta Kadambi. Synthetic generation of face videos with plethysmograph physiology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20587–20596, 2022.
- [11] Xin Liu, Yuntao Wang, Sinan Xie, Xiaoyu Zhang, Zixian Ma, Daniel McDuff, and Shwetak Patel. Mobilephys: Personalized mobile camera-based contactless physiological sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(1), mar 2022.
- [12] Xin Liu, Ziheng Jiang, Josh Fromm, Xuhai Xu, Shwetak Patel, and Daniel McDuff. Meta-phys: few-shot adaptation for non-contact physiological measurement. In *Proceedings of the conference on health, inference, and learning*, pages 154–163, 2021.
- [13] Xin Liu, Mingchuan Zhang, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. Federated remote physiological measurement with imperfect data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2155–2164, 2022.
- [14] Zhaodong Sun and Xiaobai Li. Contrast-phys: Unsupervised video-based remote physiological measurement via spatiotemporal contrast. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 492–510. Springer, 2022.
- [15] John Gideon and Simon Stent. The way to my heart is through contrastive learning: Remote photoplethysmography from unlabelled video. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3995–4004, 2021.
- [16] Jeremy Speth, Nathan Vance, Patrick Flynn, and Adam Czajka. Non-contrastive unsupervised learning of physiological signals from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14464–14474, 2023.
- [17] Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, and Daniel McDuff. Simper: Simple self-supervised learning of periodic targets. *arXiv preprint arXiv:2210.03115*, 2022.
- [18] Daniel McDuff and Ethan Blackford. iphys: An open non-contact imaging-based physiological measurement toolbox. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6521–6524. IEEE, 2019.
- [19] Giuseppe Boccignone, Donatello Conte, Vittorio Cuculo, Alessandro d’Amelio, Giuliano Grossi, and Raffaella Lanzarotti. An open framework for remote-ppg methods and their assessment. *IEEE Access*, 8:216083–216103, 2020.
- [20] Christian Pilz. On the vector space in photoplethysmography imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [21] Daniel McDuff, Sarah Gontarek, and Rosalind W. Picard. Remote Detection of Photoplethysmographic Systolic and Diastolic Peaks Using a Digital Camera. *IEEE Transactions on Biomedical Engineering*, 61(12):2948–2954, December 2014.
- [22] Giuseppe Boccignone, Donatello Conte, Vittorio Cuculo, Alessandro D’Amelio, Giuliano Grossi, Raffaella Lanzarotti, and Edoardo Mortara. pyvhr: a python framework for remote photoplethysmography. *PeerJ Computer Science*, 8:e929, 2022.
- [23] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2019.
- [24] Ronny Stricker, Steffen Müller, and Horst-Michael Gross. Non-contact video-based pulse rate measurement on a mobile service robot. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 1056–1062. IEEE, 2014.
- [25] Daniel McDuff, Miah Wander, Xin Liu, Brian L Hill, Javier Hernandez, Jonathan Lester, and Tadas Baltrusaitis. Scamps: Synthetics for camera measurement of physiological signals. *arXiv preprint arXiv:2206.04197*, 2022.
- [26] Jiankai Tang, Kequan Chen, Yuntao Wang, Yuanchun Shi, Shwetak Patel, Daniel McDuff, and Xin Liu. Mmpd: Multi-domain mobile video physiology dataset, 2023.
- [27] Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, et al. Multimodal spontaneous emotion corpus

- for human behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, 2016.
- [28] Rita Meziati Sabour, Yannick Benezeth, Pierre De Oliveira, Julien Chappe, and Fan Yang. Ubc-phys: A multimodal database for psychophysiological studies of social stress. *IEEE Transactions on Affective Computing*, 2021.
- [29] Wim Verkrusse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Optics express*, 16(26):21434–21445, 2008.
- [30] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE transactions on biomedical engineering*, 58(1):7–11, 2010.
- [31] Gerard De Haan and Vincent Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, 2013.
- [32] Wenjin Wang, Albertus C den Brinker, Sander Stuijk, and Gerard De Haan. Algorithmic principles of remote ppg. *IEEE Transactions on Biomedical Engineering*, 64(7):1479–1491, 2016.
- [33] Gerard De Haan and Arno Van Leest. Improved motion robustness of remote-ppg by using the blood volume pulse signature. *Physiological measurement*, 35(9):1913, 2014.
- [34] Christian S Pilz, Sebastian Zaunseder, Jarek Krajewski, and Vladimir Blazek. Local group invariance for heart rate estimation from face videos in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1254–1262, 2018.
- [35] Weixuan Chen and Daniel McDuff. Deepphys: Video-based physiological measurement using convolutional attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–365, 2018.
- [36] Xin Liu, Brian Hill, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. Efficientphys: Enabling simple, fast and accurate camera-based cardiac measurement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5008–5017, 2023.
- [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.
- [38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [40] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.
- [41] Girish Narayanswamy, Yujia Liu, Yuzhe Yang, Chengqian Ma, Xin Liu, Daniel McDuff, and Shwetak Patel. Bigsmall: Efficient multi-task learning for disparate spatial and temporal physiological measurements. *arXiv preprint arXiv:2303.11573*, 2023.
- [42] Daniel McDuff, Theodore Curran, and Achuta Kadambi. Synthetic data in healthcare. *arXiv preprint arXiv:2304.03243*, 2023.
- [43] Akshay Paruchuri, Xin Liu, Yulu Pan, Shwetak Patel, Daniel McDuff, and Soumyadip Sengupta. Motion matters: Neural motion transfer for better camera physiological sensing. *arXiv preprint arXiv:2303.12059*, 2023.
- [44] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. IEEE, 2018.
- [45] Danish Contractor, Daniel McDuff, Julia Katherine Haines, Jenny Lee, Christopher Hines, Brent Hecht, Nicholas Vincent, and Hanlin Li. Behavioral use licensing for responsible ai. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 778–788, 2022.