Biomedical Signal Processing and Control 113 (2026) 108886

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

GVT2RPM: An empirical study for general video transformer adaptation to remote physiological measurement

Hao Wang [a], Euijoon Ahn [b], Andrew Joseph [c], Faraz Pathan [c,d,e], Kazuaki Negishi [e,f,g,h,i], Jinman Kim [a,*]

[a] School of Computer Science, Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia
[b] College of Science and Engineering, James Cook University, Cairns, QLD 4870, Australia
[c] Department of Cardiology, Nepean Hospital, Sydney, NSW 2747, Australia
[d] Nepean Clinical School, Faculty of Medicine and Health, The University of Sydney, Sydney, NSW 2747, Australia
[e] Sydney Medical School Nepean, Charles Perkins Centre, The University of Sydney, Sydney, NSW 2747, Australia
[f] South West Sydney Clinical Campus, UNSW Sydney, Sydney, NSW 2170, Australia
[g] Liverpool Hospital, Liverpool, NSW, 2170, Australia
[h] The Ingham Institute for Applied Medical Research, Sydney, NSW, Australia
[i] Victor Chang Cardiac Research Institute, Darlinghurst, NSW, Australia

ARTICLE INFO

Keywords:
Human facial video analysis
Remote physiological measurement

ABSTRACT

Remote physiological measurement (RPM) is an essential tool for healthcare monitoring as it enables the measurement of physiological signs, e.g., heart rate, in a remote setting, via physical wearables. Recent advancements in facial video-based RPMs have leveraged video analysis to detect photoplethysmographic (PPG) changes by learning pixel variations across frames. Transformer architectures, known for their success in natural video understanding, have also been applied to facial video-based RPM. However, existing transformer-based RPM methods often rely on RPM-specific modules, such as temporal difference convolutions and handcrafted feature maps, to capture subtle physiological signals and enhance temporal feature extraction. While these customized modules can improve performance, they lack robustness across datasets and cannot be generalized to different transformer architectures due to their high degree of customization. In this study, we demonstrate that general video transformers (GVTs) can achieve state-of-the-art performance for RPM without the need of RPM-specific modules. This approach simplifies the design process and facilitates the rapid deployment of various GVT architectures for RPM tasks. We conducted an empirical investigation into how training designs, including data preprocessing and network configurations, influence the performance of GVTs in facial video-based RPM. Furthermore, we propose practical guidelines to adapt GVTs to RPM (GVT2RPM) without the need for RPM-specific modules. Our experiments, conducted on five datasets using both intra-dataset (training and testing on the same dataset) and cross-dataset (training and testing on different datasets) settings, demonstrate that the proposed GVT2RPM guidelines outperform existing RPM-specific counterparts in most of cases. In intra-dataset experiments, it reduced mean absolute error by 5.0% (UBFC-rPPG), 35.6% (MMPD-simple), and 38.2% (MMPD). In cross-dataset experiments, it achieved reductions of 4.3% (UBFC-Phys), 13.2% (MMPD-simple), 9.5% (MMPD), and 13.4% (RLAP). The results demonstrate that our guidelines can be applied across various GVT architectures and are robust to diverse datasets, making them a promising solution for advancing RPM methodologies.

1. Introduction

Telemedicine has seen rapid growth in recent years due to the necessity and convenience offered by remote patient care [1]. One of the key telemedicine services is remote physiological measurement (RPM) that are necessary to cater for chronic and long-term patients at the convenience of their home environment [2]. Traditional approaches for physiological measurement relied on physical contact wearables, e.g., cuff-based heart rate or blood pressure monitors, which, although practical, are limited by their dependency on continuous physical attachment and can be cumbersome for long-term monitoring. To overcome these limitations, contactless RPM methods via remote photoplethysmography (rPPG) have garnered increased interest [3]. A key

* Corresponding author.
E-mail addresses: hao.wang2@sydney.edu.au (H. Wang), jinman.kim@sydney.edu.au (J. Kim).

https://doi.org/10.1016/j.bspc.2025.108886
Received 5 May 2025; Received in revised form 2 September 2025; Accepted 5 October 2025
Available online 22 October 2025

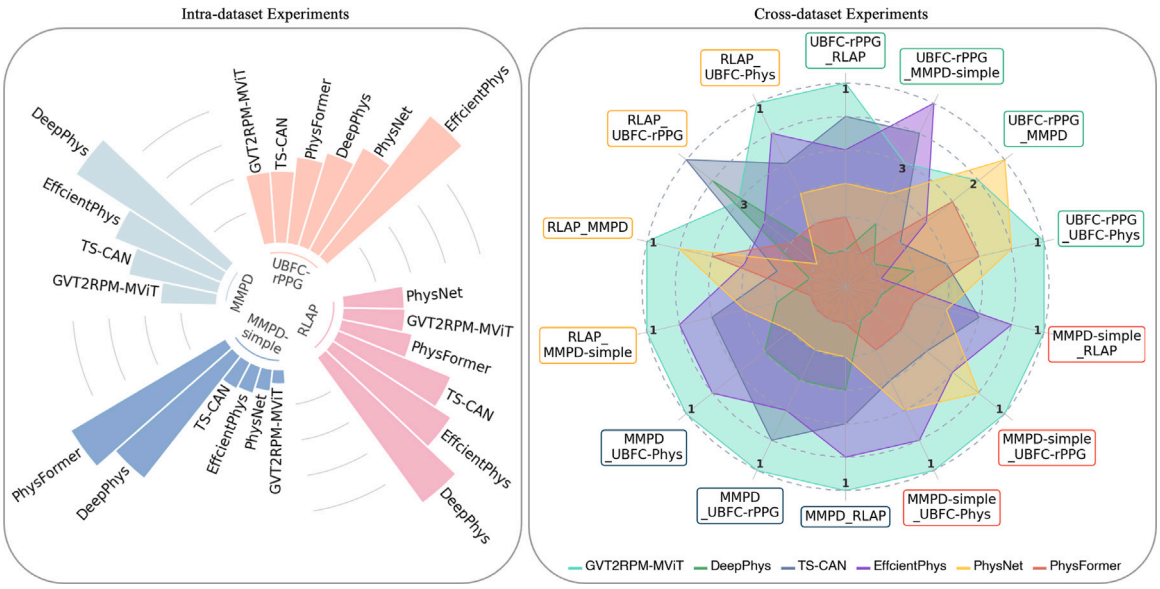


Fig. 1. Overall performance evaluations of our GVT2RPM. A general video transformer (exemplified with MViTv2) was adapted for remote physiological measurement (GVT2RPM-MViT), compared to five other state-of-the-art methods. The left graph shows the Mean Absolute Errors (MAEs) obtained in intra-dataset experiments where the training and testing sets were from the same dataset. A shorter bar means a better result. The right graph shows the ranking (based on their MAEs) of the methods in cross-dataset experiments where the training and testing sets were from different datasets. Box labels represent the name of training and testing datasets (separated by underscore).

advantage of rPPG is that it only requires a simple video camera, e.g., a smartphone camera [4]. From the acquired video, rPPG signals are derived from light changes reflected from the skin caused by the Blood Volume Pulse (BVP) [5] that are evident within the video frames, which can then be used to extract physiological parameters, such as Blood Pressure (BP) [6], Atrial Fibrillation (AF) [7], and Heart Rate (HR) [8].

In early studies, video-based RPM relied on conventional machine learning techniques to detect and process rPPG signals. For example, researchers applied blind-source signal separation techniques, e.g., independent component analysis (ICA) [9] and principal component analysis (PCA) [10], to reduce noises and recover the underlying rPPG waveforms from video frames. Furthermore, Wang et al. [11] proposed to define a plane orthogonal to the color space of the skin, which eliminated specular reflections and improved the robustness of signal processing for rPPG recovery. More recently, deep learning approaches such as Convolutional Neural Networks (CNNs) have shown promising performance in image representation learning [12–14] and video understanding [15–17]. Researchers have leveraged advanced CNN architectures to improve the efficacy of video-based RPM algorithms. For instance, Qiu et al. [18] proposed to use spatial and temporal filtering to capture facial color changes combined with CNNs to extract underlying HR information. Špetlík et al. [19] designed a two-stage CNN, where an "extractor" module was applied to learn video features. Then, a "predictor" module was used to analyze learned features for prediction. Other researchers learned video frame relationships by incorporating both spatial and temporal information, i.e., Yu et al. [20] proposed to model the spatiotemporal relationships in videos using 3D CNN or 2D CNN combined with Recurrent Neural Network (RNN).

Transformer [21], an architecture proposed in Natural Language Processing (NLP), has become a common model architecture in computer vision, especially after the proposal of the Vision Transformer (ViT) [22]. Compared to CNNs, the transformer has a larger receptive field than CNNs and thus provides a better long-range dependency. When applied to videos, the transformer can process long time-series data and can model temporal relationships between the video frames, which has contributed towards understanding actions and scenes in videos [23–25]. These advantages of the transformer architecture have also been applied to video-based RPM. For instance, Liu et al. [26] adopted the Swin Transformer, a transformer variant incorporating CNN multi-scale hierarchy, and converted 3D inputs to 2D feature maps for signal extraction. Similarly, Zhang et al. [27] applied the Transformer encoder to enable multi-scale and long-distance physiological feature learning.

Traditionally, transformer-based RPM methods have involved careful modifications of the original transformer architecture, such as replacing standard modules with RPM-specific ones like customized convolution kernels for calculating self-attention weights [28]. These approaches relied on the assumption that semantic differences between general video recognition tasks and the RPM task required specialized handling. While video transformers excel at modeling large variations in human anatomy for general tasks, RPM specifically focuses on capturing subtle color fluctuations from human skin [29] using a fixed camera viewpoint (resulting in consistent anatomy). These RPM-specific customizations, however, have drawbacks. They cannot be generalized to different transformer architectures and are not robust to various datasets. This limited flexibility restricts the adoption of state-of-the-art video transformers in RPM applications, preventing RPM methods from leveraging performance gains offered by advancements in general video transformers. Additionally, over-customization makes models more prone to overfitting, especially when training on smaller or less diverse datasets, leading to performance degradation in new clinical settings (supported by the inferior performance of PhysFormer in Fig. 1). Interestingly, recent evidence challenges the necessity of such specialized modifications. Studies such as Khan et al. [30] demonstrate that original video transformer architectures possess general capabilities that can be effectively adapted for different domains, such as audio signal feature extraction, despite semantic differences between audio learning and video understanding. This suggests that rather than requiring extensive RPM-specific modifications, standard video transformers might be adaptable for rPPG signal learning through more minimal adjustments, potentially offering both performance benefits and greater generalizability.

In this work, we address three key research questions: (1) Can general video transformers (GVT), originally designed for action recognition, be adapted for physiological signal extraction without architectural modifications? (2) What are the optimal data preprocessing

and network configuration strategies for adapting GVT to RPM across different datasets? and (3) How does the generalizability of adapted GVT compare to specialized RPM methods in cross-dataset settings? We refer GVT as transformer architectures designed for broad video understanding. We conducted an empirical study to adapt GVT to RPM (GVT2RPM) and proposed practical GVT2RPM guidelines for their adaptation to solve RPM challenges. This adaptation maintains the original transformer architectures, thus making it adaptable to various video transformer models and robust to different datasets and settings. Our guideline includes simple but effective strategies such as appropriate data pre-processing and additional temporal downsampling between the transformer blocks. Following our guidelines, we show steps to obtain optimal configurations for GVTs specific to datasets (in Section 4). As shown in Fig. 1, we evaluated various methods on five commonly used public datasets, including Multi-domain Mobile video Physiology Dataset (MMPD) [31], MMPD-simple [31], Remote Learning Affect and Physiologic dataset (RLAP) [32], Univ. Bourgogne Franche-Comté (UBFC)-rPPG [33], and UBFC-Phys [34] under intra- and cross-dataset settings using HR estimation to assess the quality of learned rPPG signals. Moreover, we conducted a majority voting based on empirical results in Section 5 and proposed general configurations for GVTs to achieve reasonable results on RPM in Section 6. Our GVT2RPM retains the original transformer architecture, allowing for straightforward integration with newly developed transformer models and diverse datasets. This flexibility reduces the need for extensive customization of RPM-specific modules, as GVT2RPM can seamlessly adopt updates from the latest advancements in transformer architectures.

1.1. Contributions

Our work presents three main contributions that advance the current state-of-the-arts:

- Unlike existing transformer-based RPM methods that rely heavily on customized modules (e.g., Temporal Difference Convolution in PhysFormer [28], handcrafted STMaps in RhythmNet [35]), we demonstrate that general video transformers can achieve superior performance without any RPM-specific modifications. This challenges the prevailing assumption in the field that specialized modules are necessary for capturing subtle physiological signals.
- We provide the first comprehensive empirical study establishing practical guidelines for adapting any general video transformer to RPM. Our guidelines are validated across 3 different GVT architectures (MViTv2 [36], UniFormer [25], Video Swin [24]) and 5 datasets, showing consistent improvements over baseline GVTs.
- Our approach demonstrates better cross-dataset performance compared to existing methods. For instance, PhysFormer shows degraded performance in cross-dataset settings (as shown in Fig. 1), while our GVT2RPM maintains robust performance, addressing a critical limitation in current RPM methods.

2. Related works

2.1. Transformer for general video analysis

Applying 3D CNNs [15,17] to capture the spatiotemporal relationships in the videos is intuitive. However, these methods have been constrained to the usage of short videos due to the limited receptive fields of the CNNs. In contrast, transformer [21] designed for sequential data learning can handle long-range relationships and is therefore suitable for processing time-series data. For example, Bertasius et al. [37] extended the ViT [22] architecture to process 3D volume inputs where the self-attention mechanism was applied to spatial and temporal dimensions separately. Similarly, Arnab et al. [38] proposed a transformer-based model in which video inputs were tokenized along spatial and temporal axes to produce 3D cubes, followed by a stack of transformer layers to learn spatiotemporal relationships. In contrast, rather than feeding raw cubes into the transformer, Neimark et al. [39] integrated inductive biases and applied CNNs to extract features from each frame before sending them into the transformer to model the temporal relationships. Moreover, Fan et al. [23] implemented multiscale feature hierarchies for the transformer to achieve efficient and effective video recognition, which they learned from the success of CNNs.

2.2. Transformer for remote physiological measurement

The transformer can be helpful when applied to RPM. For example, Liu et al. [26] proposed using tensor-shifted 2D convolutions [40] to generate 2D feature maps from 3D videos, which were then fed into the 2D Transformer to learn the spatiotemporal relationships. Similarly, Liu et al. [41] converted video inputs into handcrafted 2D spatiotemporal Map (STMap) representations [35], and then ViT was applied to extract underlying signal features. Instead of converting videos into 2D representations, Yu et al. [28,42] proposed a video-transformer-based architecture for rPPG signal representation learning that Temporal Difference Convolution (TDC) [43] was used for self-attention calculation capturing temporal difference features. However, two limitations are identified for the above methods: (1) converting 3D facial videos into handcrafted 2D STMap requires prior knowledge about physiology, resulting in biases, and (2) customized transformer modules for the RPM deteriorate the model generalizability and hinder sharing the advancements from general video recognition.

3. General Video Transformers (GVTs)

Due to the additional time dimension in video inputs, learning the temporal dependencies among the video frames is essential for video understanding. Initially, researchers [37,38] applied self-attention along the temporal axis to learn spatiotemporal relationships. This design style of transformer evolved into a hybrid structure consisting of CNN and transformer [23–25]. Specifically, a standard scheme of these methods is to integrate multiscale hierarchy in the CNN into the transformer achieving an optimal speed-accuracy trade-off. We refer this architecture as General Video Transformers (GVTs). In practice, the GVTs consists of five sequential stages as shown in Fig. 2. Firstly, the patchify stem is applied to split inputs into space–time cubes for later self-attention operations. Then, positional encodings are added for each cube. Afterward, cubes are fed into four stages sequentially to extract spatiotemporal features. Each stage contains multiple blocks consisting of transformer encoder or CNN block structures. The number of blocks in each stage depends on different model designs, but the feature map dimensions inside each stage remain the same. Therefore, the multiscale hierarchy only happens during the transition of stages and is agnostic to different hybrid video transformers.

4. GVT2RPM guidelines: Exemplified with MViTv2

This section describes our guidelines for adapting the GVT to RPM as shown in Fig. 2. We used the recent video transformer of Multiscale Vision Transformer v2 (MViTv2) [36] as our baseline. Starting from the MViTv2's standard settings, we show the changes in its performance from our adaptation in Fig. 3.

Experiment Settings. Due to the differences between general video recognition and RPM, the training strategies can vary. To make experiments consistent and reproducible, we integrated the official MViTv2 implementation[1] with rPPG-Toolbox [44] to benchmark algorithms and

[1] https://github.com/facebookresearch/SlowFast/tree/main/projects/mvitv2

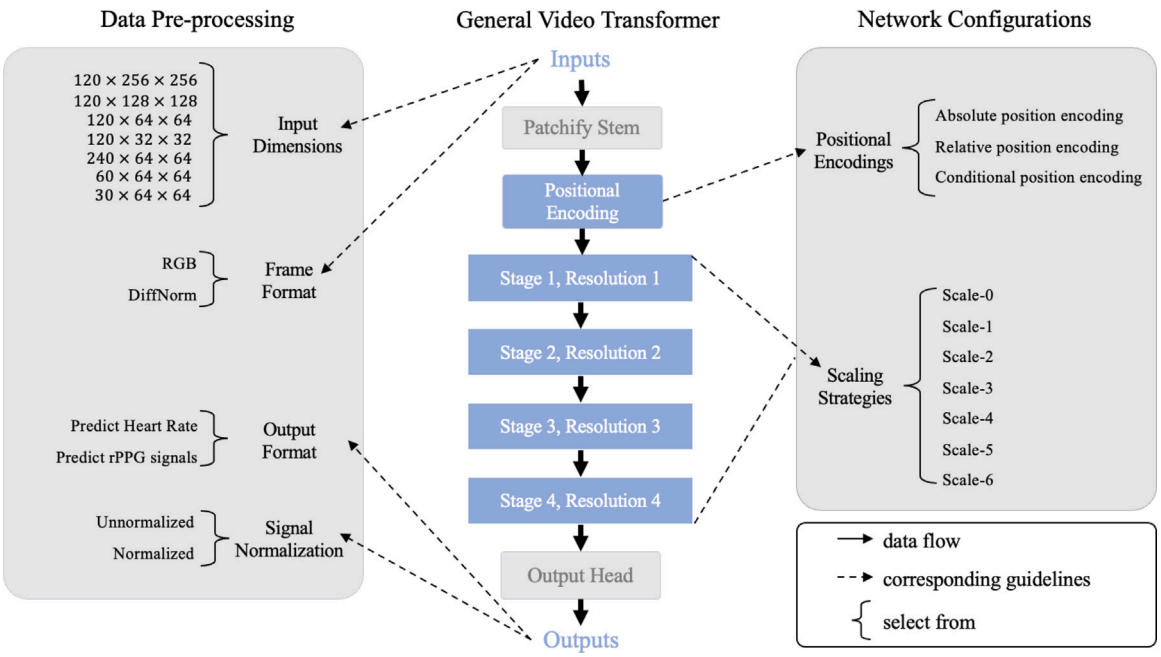


Fig. 2. Overview of our proposed guidelines for adapting general video transformer to remote physiological measurement. We used blue color to highlight the designs that could affect final performance. The parameters of each guideline are listed within the bracket and are selected based on empirical results. The decision for each design is made sequentially using a greedy algorithm, which selects the best available option at each step without revisiting previous choices. For each decision, we choose the setting that yields the lowest MAE and fix it before optimizing the next factor. For example, when evaluating input dimensions, 120 × 64 × 64 produced the minimum MAE, so it was chosen and fixed for all subsequent experiments.

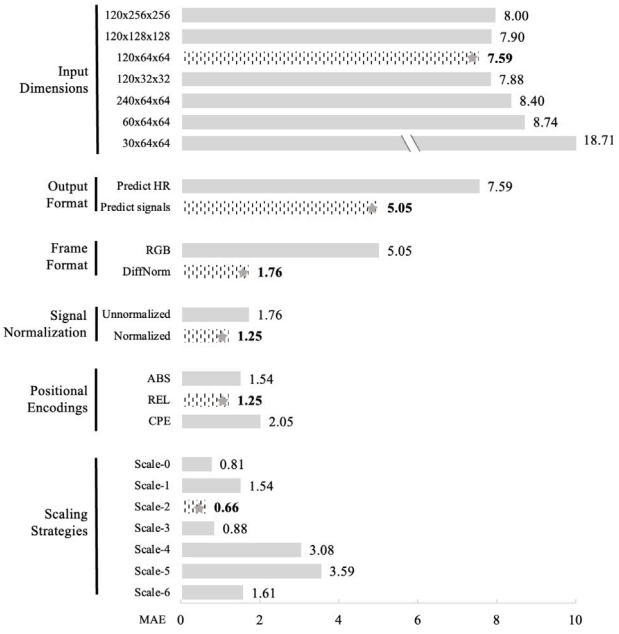


Fig. 3. Experiment results under MMPD-simple intra-dataset setting by exploring the adaption from MViTv2 to GVT2RPM-MViT. We used Mean Absolute Error (MAE) as the metric, computed by comparing the model's predicted heart rates with the ground-truth heart rates for each video clip across the entire dataset.

run experiments. We used the PyTorch [45] library and kept most default training strategies in the rPPG-Toolbox: the batch size was set to 4, and the optimizer was AdamW [46]. We extended the number of epochs to 50, modified the learning rate to 1e−3, and removed the learning rate scheduler to reduce hyperparameters. For simplicity, we used the MViTv2-S[2] as the backbone and kept the hyperparameters unchanged unless specified. The model performance was evaluated under the intra-dataset setting (train/validation/test ratio of 7:1:2, split based on subjects) on MMPD-simple [31], and the Mean Absolute Error (MAE) was used as the metric.

Our guidelines consisted of two parts: (1) data pre-processing with four sub-parts: input dimensions, output format, frame format, and signal normalization, and (2) network configuration with two sub-parts: positional encodings and scaling strategies. Following each part sequentially, the model was adapted to RPM. The choice of configuration in each part is based on a greedy algorithm, which selects the best option at the current step and does not revisit earlier configurations. We began with the default setting of input dimension with '120 × 64 × 64', output format with 'HR values', frame format - 'RGB', signal normalization - 'Disabled', positional encoding - REL, and scaling strategy with Scale-0. The first step of the guideline is to determine the input dimension of video clips. As shown in Fig. 3, an input dimension of 120 × 64 × 64 achieved the lowest MAE and was therefore fixed for subsequent experiments. Next, Fig. 3 shows that the results of using signals were better than using HR values, producing lower MAE, and thus we changed to use rPPG signals as labels. At this stage, the model design choices were: (1) an input dimension of 120 × 64 × 64 and (2) using signals as labels (changed from using HR in the default setting). The remaining steps follow the same greedy selection process.

4.1. Data pre-processing

4.1.1. Input dimensions

The input dimensions of video recognition are generally 16 × 224 × 224 (Frame length T× Height H× Width W), which is different in the RPM task. Therefore, these two tasks have opposite biases to the spatial and temporal information. In general video recognition, models need more spatial details to detect objects within the video. Still, they need

[2] Later use of MViTv2 refers to this model size unless specified.

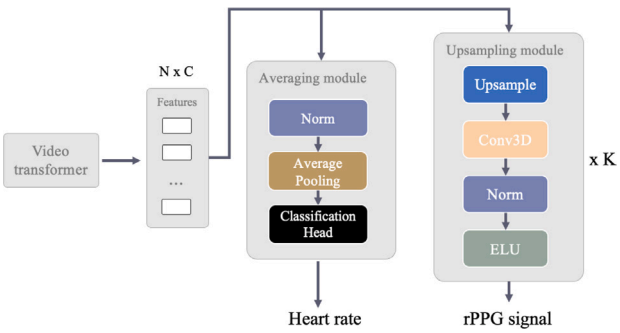


Fig. 4. Design of averaging and upsampling modules. For heart rate output, the extracted features are averaged and passed to the classification head for prediction. For signal output, the features are upsampled to generate continuous signal predictions instead of being averaged.

sparse time-related information (e.g., key frames) to define an action (such as a tennis hit's beginning, middle, and ending moments). In contrast, RPM requires dense temporal information to capture continuous rPPG signal features. The spatial information, however, is less critical, providing facial semantics and containing potential noises disturbing the training process [47,48]. Therefore, we tested a set of spatial dimensions {256, 128, 64, 32} where numbers are powers of 2 and a set of temporal dimensions {240, 120, 60, 30} where numbers are multiples of 30, which is the standard setting of Frames Per Second (FPS) for RPM. As shown in Fig. 3, we first fixed the temporal dimension to 120 since it is close to the common setting of existing RPM methods and found that spatial dimension 64 achieved the best result with an MAE of 7.59. We then fixed the spatial dimension to 64 and found that the temporal dimension 120 performed the best. *This suggests that compared with general video recognition, RPM requires a smaller frame size to reduce environmental noises and a longer clip length for enriched signal features.*

4.1.2. Output format

For video classification, the final predictions are made by either the class (CLS) token [22] or by averaging output tokens from the last transformer block and applying a classification head [36]. In RPM, outputs can be either rPPG signals or HR values derived from the rPPG signals. When the output format is HR, we can use the averaging module to make predictions without modifications. When predicting continuous signals, we appended an upsampling module to map learned features to signals . In detail, after the last transformer block, we added K of upsampling modules depending on the feature map dimensions and target signals. We show the design of upsampling module in Fig. 4. Each upsampling module consists of a nearest upsampling layer, a 3D convolutional layer, a 3D batch normalization layer, and an Exponential Linear Unit (ELU) activation layer. The scaling factor of the upsampling layer was set to (2, 1, 1). The 3D convolutional layer kept the input and output dimensions the same with a kernel size of (3, 1, 1), stride of 1, and padding of (1, 0, 0). *In the experiment, using rPPG signals as ground truth reduces the MAE to 5.05, showing that signals contain more information than HR values.*

4.1.3. Video frame format

Using raw RGB video frames as inputs is common in video recognition. However, based on the skin reflection model [11], RGB inputs can be sub-optimal, affecting algorithm performances due to the reflection noises resulting from the light source and skin tone of subjects. Therefore, calculating Differences of Normalized frames (DiffNorm) [47], which minimizes the RGB input constraints, has become a popular data pre-processing to capture underlying rPPG signals under various illumination conditions. *After applying DiffNorm to raw inputs, the MAE dropped from 5.05 to 1.76 (see Fig. 3), indicating the effectiveness of DiffNorm in reducing inherent noises in RGB-format videos.*

4.1.4. Signal normalization

Normalizing signals into the same scale can help stabilize the gradient descent steps and improve the model convergence rate in most cases [49]. Standardization, which transforms values to have a mean of 0 and a standard deviation of 1, is a prevalent normalization technique, assuming the data follows a Gaussian distribution [50]. However, this assumption can be invalid for some datasets and thus hinder the model training. *In our experiment, signal normalization reduced the MAE by 0.51 to 1.25 (see Fig. 3).*

4.2. Network configurations

4.2.1. Position encodings

In contrast to CNNs, which inherently contain positional information by the sliding window operation, the transformer processes all input tokens in parallel without referring to the order or positions. Understanding positional information is essential in vision recognition, and it helps to learn high-level semantic meanings like relationships between objects [51]. In our studies, we evaluated three different position encodings, including absolute position encodings (ABS) [23], relative position encodings (REL) [36], and conditional position encodings (CPE) [52]. Although ViT [22] speculated that ABS and REL have no differences in image classification, MViTv2 [36] found that REL can achieve better performances in video recognition. Recently, CPE was proposed to integrate translation equivalence into the vision transformer to improve performance. Since CPE was initially proposed for 2D images, we extended it for 3D video inputs by replacing the 2D CNN layers with 3D CNN. *In this experiment, REL performed better than the other two position encodings with an MAE of 1.25 (see Fig. 3).*

4.2.2. Scaling strategies

The key idea of modern GVTs is to integrate multiscale feature hierarchies in CNNs with the transformer architecture. This is implemented by reducing the spatial resolution of feature maps and increasing the channel capacity at certain stages. With prior knowledge about rPPG signals, RPM methods are required to extract dense temporal signals from facial videos with more frames than general video analysis (e.g., 120 v.s. 16 frames). Therefore, the default scaling strategy in GVTs, which only downsample over spatial dimensions after the patchify stem, can be suboptimal for RPM. We designed and experimented with different scaling strategies to investigate how space–time hierarchies affect model performances. In addition to pooling spatial resolutions only (Scale-0), we implemented pooling over temporal resolution to emphasize time hierarchy efficacy in RPM. As shown in Fig. 5, Scale-1, Scale-2, and Scale-3 involve temporal downsampling at Stage 1, Stage 2, and Stage 3, respectively. Meanwhile, Scale-4, Scale-5, and Scale-6 employ more aggressive temporal scaling, reducing the temporal resolution at two stages through different combinations. *We find that introducing appropriate temporal downsampling is beneficial; in our case, Scale-2 achieved the lowest MAE of 0.66 (see Fig. 3).*

Summary. Following the proposed guidelines, we derived optimal configurations for adapting MViTv2 to RPM (GVT2RPM-MViT) on the MMPD-simple dataset: input dimension of 120 × 64 × 64, using rPPG signals as labels, applying DiffNorm for frames representation, normalizing signals, using REL position encoding, and applying Scale-2 strategy. This adaptation largely improved performance, reducing the MAE from 7.59 to 0.66, a 91.3% decrease, compared to the standard MViTv2 with the same input dimensions (see Fig. 3).

5. Empirical evaluations on different datasets and settings

Following the GVT2RPM guidelines in Section 4, we exemplified how to adapt a GVT (e.g., MViTv2) to RPM and find optimal configurations for a specific dataset without using RPM-specific modules. In this section, we evaluate the robustness of GVT2RPM through intra-dataset experiments on three additional datasets and cross-dataset experiments across five datasets. Based on the empirical results, we provide practical insights for selecting optimal GVT configurations tailored to different datasets.

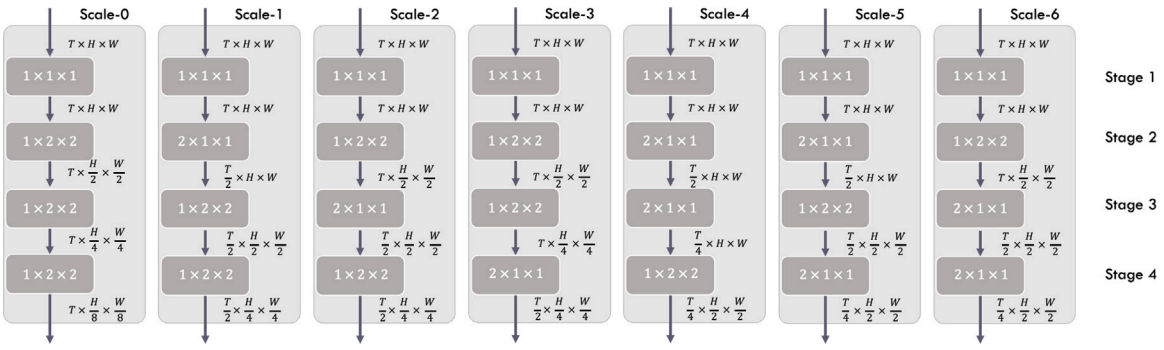


Fig. 5. Details of scaling strategies. The default scaling strategy in GVTs, Scale-0, applies downsampling only to spatial dimensions after the patchify stem. To introduce space–time hierarchies, we developed Scale-1, Scale-2, and Scale-3 by incorporating temporal downsampling at Stages 1, 2, and 3, respectively. Furthermore, we created Scale-4, Scale-5, and Scale-6 by adding temporal downsampling across two stages with varying combinations.

Table 1
Results of adapting MViTv2 to GVT2RPM-MViT on MMPD-simple, MMPD, RLAP, and UBFC-rPPG under intra-dataset settings. MAE is used as the metric. The best combination of video frame format and signal normalization has a green background, the best positional encoding has a blue background, and the best scaling strategy has a yellow background.

Datasets	Raw input		DiffNorm input		Positional encoding			Scaling strategies						
	w/o signal norm	with signal norm	w/o signal norm	with signal norm	ABS	REL	CPE	Scale-0	Scale-1	Scale-2	Scale-3	Scale-4	Scale-5	Scale-6
MMPD-simple	5.05	4.03	1.76	**1.25**	1.54	**1.25**	2.05	0.81	1.54	**0.66**	0.88	3.08	3.59	1.61
MMPD	13.06	7.23	**7.22**	7.72	7.3	**7.22**	8.83	8.65	7.21	8.67	8.37	7.69	7.04	**6.83**
RLAP	17.36	1.69	1.67	**1.48**	1.87	**1.48**	2.11	1.7	1.69	**1.38**	1.45	1.54	1.82	1.74
UBFC-rPPG	3.03	3.91	2.93	**2.83**	**2.14**	2.83	2.15	2.15	1.76	2.25	1.66	1.95	**1.56**	2.93

5.1. Datasets

- MMPD [31]: This dataset contains 660 videos recorded by a Samsung Galaxy S22 Ultra mobile phone at 30 FPS with a resolution of 1280 × 720 and compressed to 320 × 240 stored in H.264 format. An HKG-07C+ oximeter records the ground truth PGG signals. Videos are recorded under four lighting conditions, motions, and skin tones. This dataset contain 33 subjects with 1,188,000 frames.
- MMPD-Simple [31]: Due to the difficulty of the original MMPD, authors created a subset to contain videos with stationary, skin tone type 3, and artificial light conditions. It has 49 videos with 132,300 frames.
- RLAP [32]: This dataset contains 754 videos recorded by a Logitech C920c webcam at 30 FPS with a resolution of 1920 × 1080 stored in MJPG format. A CMS50E transmissive pulse oximeter records the ground truth PPG signals. During video recording, subjects completed tasks or watched videos under different lighting conditions. This dataset contain 58 subjects with 3,530,000 frames.
- UBFC-rPPG [33]: This dataset contains 46 videos recorded by a Logitech C920 HD Pro webcam at 30 FPS with a resolution of 640 × 480 in uncompressed 8-bit RGB format. A CMS50E transmissive pulse oximeter records corresponding PPG signals. The recording is conducted indoors with sufficient sunlight and artificial illumination. This dataset contain 42 subjects with 57,420 frames.
- UBFC-Phys [34]: This dataset contains 168 videos recorded by an EO-23121C RGB digital camera at 35 FPS with a resolution of 1024 × 1024 stored in MJPG format. The underlying BVP signals were recorded by the Empatica E4 wristband. The collection is conducted with three tasks with significant amounts of unconstrained motion under static lighting conditions. This dataset contain 56 subjects with 1,048,320 frames.

5.2. Experiment settings

We conducted two experiment protocols, including intra-dataset and cross-dataset experiments. For the intra-dataset experiments, models were trained, validated, and tested on the same dataset with a split ratio of 7:1:2. For the cross-dataset experiments, models were trained and validated on the same dataset with a split ratio of 8:2, and then tested on another dataset.

The training process was the same as in Section 4. It was consistent for intra-dataset and cross-dataset experiments, except that we fixed the input dimensions to 120 × 64 × 64 and the output format to rPPG signals, as this combination consistently had better performances than other combinations. We evaluated the model performance based on the metric MAE.

The exploration of model designs consisted of 3 parts: video frame format and signal normalization, positional encodings, and scaling strategies. Each part choice was based on the greedy algorithm.

5.3. Intra-dataset experiments

We conducted intra-dataset experiments on MMPD-simple, MMPD, RLAP, and UBFC-rPPG. Table 1 summarizes the performance of different designs of GVT2RPM-MViT on intra-dataset experiments. There are some findings we have identified:

5.3.1. DiffNorm helps in most cases.

Our results suggest that videos pre-processed by the DiffNorm always performed better, compared to using raw RGB inputs. Considering the case of using raw signals (w/o signal norm), DiffNorm reduced MAE over half for MMPD-simple and RLAP, 44.7% for MMPD, and 3.3% for UBFC-rPPG. This demonstrates that DiffNorm was key to adapting a GVT for RPM. It amplifies the underlying rPPG signals by suppressing the motion and illumination noises in the raw RGB videos and enforces the transformer to focus on subtle pixel variations instead of human anatomy.

Table 2
Results of adapting MViTv2 to GVT2RPM-MViT on MMPD-simple, MMPD, RLAP, UBFC-rPPG, and UBFC-Phys under cross-dataset settings. MAE is used as the evaluation metric. The best combination of video frame format and signal normalization has a green background, the best positional encoding has a blue background, and the best scaling strategy has a yellow background.

Train dataset	Test dataset	Raw input		DiffNorm input		Positional encoding			Scaling strategies						
		w/o signal norm	with signal norm	w/o signal norm	with signal norm	ABS	REL	CPE	Scale-0	Scale-1	Scale-2	Scale-3	Scale-4	Scale-5	Scale-6
MMPD-simple	UBFC-rPPG	28.17	27.71	**8.14**	9.12	7.3	8.14	**1.46**	**6.11**	13.81	12.72	7.76	25.3	23.14	21.66
	UBFC-Phys	12.42	11.13	**6.04**	7.14	**5.09**	6.04	7.04	**5.47**	5.92	5.92	5.89	7.5	6.28	7.28
	RLAP	7.68	8.82	**5.23**	5.31	5.81	5.23	**3.7**	5.78	3.27	**3.27**	4.58	6.44	4.99	4.96
MMPD	UBFC-rPPG	20.57	15.95	**5.75**	8.06	2.78	5.75	**2.49**	**2.2**	3.87	4.46	3.98	8.56	7.47	9.14
	UBFC-Phys	10.8	24.68	**5.49**	10.18	**5.16**	5.49	6.24	**4.5**	5.69	6.07	6.05	5.26	4.95	7.9
	RLAP	20.19	15.48	**3.42**	7.16	**3.4**	3.42	3.76	**3.98**	5.23	4.75	4.57	4.04	5.14	4.76
RLAP	MMPD-simple	18.07	4.71	3.23	**1.92**	**0.79**	1.92	1.38	**0.97**	2.31	3.44	1.17	2.59	2.6	4.14
	MMPD	13.48	11.75	10.03	**9.02**	9.95	**9.02**	9.75	9.2	9.44	**8.58**	9.72	8.86	9.39	9.7
	UBFC-rPPG	19.48	12.74	6.36	**5.57**	2.05	5.57	**1.48**	2.36	5.44	4.33	**1.9**	5.57	5.61	4.04
	UBFC-Phys	10.97	4.96	4.57	**4.32**	**4.19**	4.32	4.31	4.31	4.68	4.9	**4.17**	4.44	4.76	4.49
UBFC-rPPG	MMPD-simple	16.8	22.24	1.96	**1.87**	3.14	**1.87**	2.55	**2.73**	4.34	5.94	4.39	10.6	7.32	5.54
	MMPD	14.12	17.31	12.2	**11.5**	12.47	**11.5**	12.67	11.71	**11.28**	11.71	13.16	12.02	12.02	11.45
	UBFC-Phys	6.46	6.2	**4.49**	4.75	**4.36**	4.49	5.3	4.72	5.31	4.78	**4.36**	5.01	4.99	5.11
	RLAP	6.63	7.41	**3.01**	3.27	3.49	**3.01**	3.49	3.32	3.45	3.26	**3.07**	3.91	3.88	3.35

5.3.2. Signal normalization helps in simple scenarios.

We observe that signal normalization helps when the dataset contains relatively simple settings, e.g., non-rigid movement and constant and sufficient illumination, such that the MAE was reduced from 1.76 to 1.25 in MMPD-simple, from 1.67 to 1.48 in RLAP, and from 2.93 to 2.83 in UBFC-rPPG. MMPD, having rigid head motions, various skin tones, and changing lighting conditions, can cause many outliers and break Gaussian distribution, thus making it incompatible with signal normalization.

5.3.3. Relative positional encoding is robust in most cases.

The REL obtained lower MAEs on MMPD-simple, MMPD, and RLAP than the other two positional encodings. Except for UBFC-rPPG, the ABS achieved a lower MAE of 2.14.

5.3.4. Appropriate temporal hierarchy helps signal learning.

We find that adding temporal downsampling between transformer blocks assisted better understanding of signals such that the MAE was reduced from 0.81 to 0.66 in MMPD-simple, from 8.65 to 6.83 in MMPD, from 1.7 to 1.38 in RLAP, and from 2.14 to 1.56 in UBFC-rPPG. These results suggest that temporal downsampling enhances the model's efficiency in processing dense temporal information over longer intervals. It allows the model to focus on critical signal features within a consistent context, particularly when the training and testing datasets exhibit similar temporal dynamics and visual characteristics.

5.3.5. Summary

Based on the empirical results of intra-dataset experiments, we conducted a majority voting and proposed general configurations for GVTs: (1) for data pre-processing, the input clip dimension was $120 \times 64 \times 64$, rPPG signals were used for outputs, and each frame was pre-processed by DiffNorm; (2) depending on dataset complexity, output signals were normalized for simple scenarios; (3) for network configurations, REL positional encoding was used, and the scaling strategy was set to Scale-2.

5.4. Cross-dataset experiments

We conducted cross-dataset experiments on MMPD-simple, MMPD, UBFC-rPPG, UBFC-Phys, and RLAP. Table 2 shows the performance of different designs of GVT2RPM-MViT on cross-dataset experiments. Since MMPD-Simple is a subset of MMPD, we excluded them due to the data leakage when training on MMPD-Simple and testing on MMPD, and vice versa. The choices of designs have different effects compared with the intra-dataset setting, and we conclude:

5.4.1. DiffNorm significantly improves transfer learning.

Consistent with the intra-dataset experiments, DiffNorm proved advantageous for transfer learning. We observed that applying DiffNorm reduced MAEs by an average of 55.6% across all cases when using raw signals, and by 43.9% when using normalized signals.

5.4.2. The efficacy of signal normalization depends on the training dataset.

We noticed that signal normalization can hinder model learning when we trained on MMPD-simple and MMPD. It suggests that using raw signals is a better choice when transferring from low-quality video datasets (H.264 compressed) to higher-quality datasets (MJPG compressed or uncompressed). In contrast, when training on the high-quality RLAP, normalized signals are better for the transfer learning.

5.4.3. Positional encoding can be selected based on the target dataset.

Unlike intra-dataset experiments where REL outperformed in most cases, the cross-dataset experiments revealed dataset-specific preferences. CPE achieved better performance when the target dataset was UBFC-rPPG, reducing MAEs by an average of 70.7% compared to REL. Similarly, ABS excelled for UBFC-Phys, lowering MAEs by an average of 6.91% when replacing REL.

5.4.4. Spatial hierarchy is more robust for transfer learning.

We observe that half of the experiments showed better performances with Scale-0, the default GVT setting where no temporal downsampling is applied. The other half performed better with Scale-1, Scale-2, or Scale-3, which apply a single temporal downsampling between transformer blocks. However, applying multiple temporal downsampling steps (Scale-4, Scale-5, and Scale-6) hindered the model's ability to learn robust signal features. This indicates that excessive temporal downsampling may obscure subtle temporal variations essential for RPM task, especially those affected by dataset-specific factors such as lighting conditions, head movements, and camera settings.

5.4.5. Summary

Based on the empirical results from cross-dataset experiments, we proposed general configurations for GVTs in cross-dataset scenarios: (1) for data pre-processing, the input clip dimension was $120 \times 64 \times 64$, rPPG signals were used for outputs, and each frame was pre-processed by DiffNorm; (2) depending on dataset quality, output signals were normalized when training on high-quality datasets; (3) the choice of position encoding was determined by the target dataset, such that CPE for UBFC-rPPG, ABS for UBFC-Phys, and REL for all other cases; (4) the scaling strategy was fixed to Scale-0.

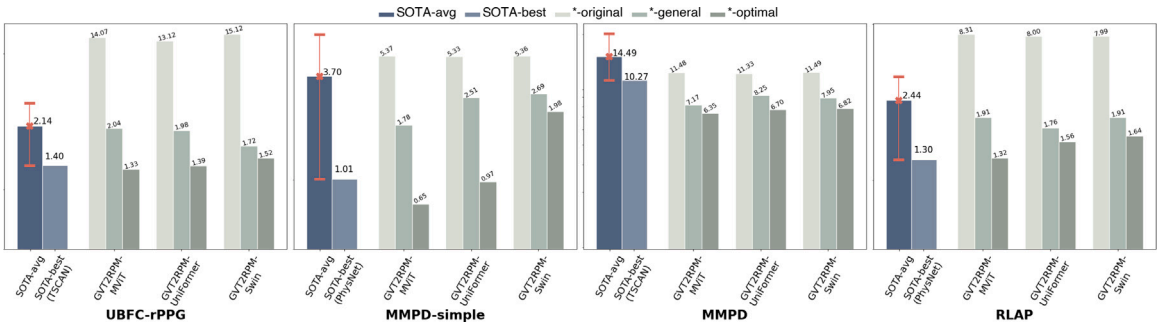


Fig. 6. Intra-dataset experiment results on MMPD-simple, MMPD, RLAP, and UBFC-rPPG. We evaluated five SOTA methods for RPM. Their averaged results are denoted by SOTA-avg with red error bar. The best performing method is denoted by SOTA-best. Also, we tested three GVTs, including MViTv2, UniFormer, and Video Swin. Based on our empirical results in Section 5, we constructed GVT2RPM-MViT-general, GVT2RPM-UniFormer-general, and GVT2RPM-Swin-general. Following Section 4, we further optimized them into GVT2RPM-MViT-optimal, GVT2RPM-UniFormer-optimal, and GVT2RPM-Swin-optimal. We used MAE as the metric.

6. Evaluations of different GVTs

This section demonstrates the application of our GVT2RPM to multiple GVTs where we employed the recent SOTA GVT of UniFormer [25], and Video Swin [24], in addition to the MViTv2 [36] from Section 4. We conducted intra-dataset experiments to evaluate their performance on MMPD-simple, MMPD, RLAP, and UBFC-rPPG. For fair comparisons, we applied 3-fold cross-validation except for RLAP, where the official split [32] was used. All models were trained from scratch.

We applied the *general configurations* in Section 5.3.5 to the above GVTs and adapted them to GVT2RPM-MViT-general, GVT2RPM-UniFormer-general, and GVT2RPM-Swin-general. Additionally, each model was optimized following the proposed guidelines in Section 4 to obtain the *optimal configurations*. Therefore, we have GVT2RPM-MViT-optimal, GVT2RPM-UniFormer-optimal, and GVT2RPM-Swin-optimal (configuration details in the Appendix A). To compare, we trained SOTA RPM methods, including DeepPhys [47], PhysNet [20], TSCAN [53], EfficientPhys [26], and PhysFormer [28], using rPPG-toolbox with default settings and averaged their performance as the baseline, named SOTA-avg. The best-performing method was denoted as SOTA-best.

Experimental results are shown in Fig. 6. We observe that applying our general configurations successfully adapted GVTs to RPM, where all three GVT2RPM-*-general methods achieved better MAEs than the SOTA-avg in four datasets. In contrast, naively using GVTs for RPM with their original versions (GVT2RPM-*-original), as expected, performed worse than the SOTA-avg except with the MMPD. This indicates that our GVT2RPM is generalizable to different GVTs and robust to various datasets.

Moreover, optimizing GVTs using our guidelines delivered better results, competing favorably with RPM-specific SOTA methods. For instance, GVT2RPM-MViT-optimal obtained better results than the other two GVTs and outperformed the SOTA-best in UBFC-rPPG, MMPD-simple, and MMPD datasets. Notably, the superior performance of GVT2RPM-MViT aligns with trends in general video tasks, where MViTv2, with practical enhancements such as residual connections and Key-Value pooling, demonstrated better video representations compared to UniFormer and Video Swin. Furthermore, we observe that the optimization of MViTv2 from its original version yielded greater improvements than the other two GVTs, with average improvements across four datasets of 76.81% for MViTv2, 73.14% for UniFormer, and 68.28% for Video Swin.

Across the four datasets, all methods showed reduced performance on MMPD, likely due to its more challenging scenarios, such as rigid patient face movements, varying lighting conditions, and diverse skin tones, compared to the other datasets. It is interesting to note that in this complex dataset, all GVT2RPM-*-origin models outperformed the SOTA-avg, highlighting the superior capability of GVTs to handle complex data. Furthermore, our GVT2RPM guidelines improved their performance, with both GVT2RPM-*-general and GVT2RPM-*-optimal achieving lower MAEs than the SOTA-best.

Table 3
The computational cost of scaling strategies. The efficiency was measured by the total number of model parameters with the unit of Million (M), Floating Point Operations Per Second (FLOPs) with the unit of Giga (G), and the model inference speed with the unit of frames per sec. (frame/sec.).

Scaling strategies	#Params (M)	FLOPs (G)	Throughput (frame/sec.)
Scale-0 (Original)	35,997,601	115.6	3644
Scale-1	37,712,545	217.5	2102
Scale-2	37,717,921	167.9	2874
Scale-3	37,764,385	123.5	3740
Scale-4	39,485,089	383.3	1289
Scale-5	39,482,977	244.6	2071
Scale-6	39,488,353	195.1	2812

7. Computational cost

We evaluated the computational costs of different scaling strategies via the total number of model parameters with the unit of Million (M), Floating Point Operations Per Second (FLOPs) with the unit of Giga (G), and the inference speed with the unit of frames per sec. (frame/sec.). All algorithms were run on a machine with Intel(R) Core(TM) i9-9900 K CPU and a single Nvidia GeForce RTX 4090 24G. The input data came from the MMPD-simple dataset and had the size of $1 \times 3 \times 120 \times 64 \times 64$ representing batch size × channels × frames × height × width. As shown in Table 3, compared with the original GVT scaling strategy Scale-0, due to aggressive temporal scaling in the early stages, the strategy Scale-4 had the worst efficiency achieving only 1/3 of inference speed. In contrast, the strategy Scale-2, which is the best strategy for intra-dataset experiments, achieved 18.5% better MAE with the cost of 21.1% lower inference speed compared to the original GVT scaling strategy. It shows that there exists an optimal trade-off between model accuracy and computational efficiency when applying temporal downsampling. We found that Scale-3 achieved slightly better inference speed (3740 frame/sec.) than the original Scale-0 (3644 frame/sec.) while maintaining similar FLOPs, suggesting that strategic placement of temporal downsampling at later stages can improve inference speed without significantly increasing computational burden. These findings demonstrate that the distribution of temporal downsampling operations across network stages can impact both model performance and efficiency, with early aggressive downsampling (as in Scale-4) largely reducing throughput, while more balanced approaches (like Scale-2 and Scale-3) offer more favorable accuracy-efficiency trade-offs for the RPM.

8. Discussion

Our experimental design validates our core hypothesis that GVTs can be successfully adapted for RPM through systematic configuration optimization rather than architectural redesign. Regarding RQ1: Our key finding demonstrates that unmodified GVT architectures achieve 77% performance improvements over baseline configurations, with GVT2RPM-MViT-optimal outperforming specialized methods on three datasets, as shown in experiments in Section 5.3. This directly validates our hypothesis and establishes our primary novelty - that RPM-specific architectural modifications (like Temporal Difference Convolution in PhysFormer) are unnecessary. The experimental evidence validates the hypothesis that appropriate data preprocessing (particularly DiffNorm) and configuration adjustments are sufficient to adapt general video understanding capabilities to physiological signal extraction tasks. Regarding RQ2: Our hypothesis was that systematic exploration of configuration parameters would reveal generalizable patterns that could be summarized into practical guidelines, despite dataset-specific variations in optimal settings. The key finding validates this hypothesis through our greedy optimization approach across six design dimensions (Section 4), which consistently identified beneficial configurations across multiple datasets. Specifically, DiffNorm preprocessing emerged as consistently beneficial for various datasets, while other parameters like temporal downsampling showed predictable context-dependent patterns (Scale-2 for intra-dataset, Scale-0 for cross-dataset scenarios). This experimental evidence supports our hypothesis that while optimal configurations vary by dataset complexity and experimental setting, underlying principles can be systematically identified and generalized. Our novelty lies in transforming the traditionally ad-hoc process of RPM method customization into a systematic empirical framework, providing the first comprehensive guidelines for adapting any GVT architecture to RPM through principled configuration optimization rather than trial-and-error approaches. Regarding RQ3: Our cross-dataset experiments (in Section 5 and Fig. 1) reveal superior generalization compared to existing methods, with consistent performance maintenance while specialized methods like PhysFormer show degraded transfer performance. The key finding is that our approach achieves 4.3–13.4% MAE reductions across different dataset combinations. This validates our hypothesis that avoiding over-specialized architectural modifications enhances robustness. Our novelty is demonstrated through the first systematic cross-dataset evaluation showing that general video understanding capabilities transfer more effectively to diverse RPM scenarios than highly customized approaches.

The convergent evidence across all three research questions supports our central hypothesis that general video transformers possess inherent capabilities for physiological signal learning that can be unlocked through systematic adaptation rather than architectural redesign.

9. Limitations

We identified three limitations in this work. First, we did not investigate the impact of patients' skin tones on performance, which introduces additional challenges for robustness due to variations in skin tone reflections. We will explore color normalization and video synthesis techniques in future research. Second, we used a smaller version of the video transformer with fewer parameters compared to larger variants. While this smaller model offers better efficiency and ease of training, it yields slightly lower performance. In future work, we will assess if our guidelines generalize to larger models. Given that our GVT2RPM guidelines has no restrictions on the GVT size, we expect better results with larger GVTs. Lastly, the optimal configurations in our experiments were selected manually based on our guidelines. Automating this process, for instance, by adopting approaches similar to nnUNet [54], could streamline configuration selection through interdependent rules and empirical decision-making.

10. Conclusions

In this paper, we conducted empirical research for adapting GVTs to the RPM. We demonstrated that GVT2RPM adapted GVTs can achieve reasonable performance without relying on RPM-specific modules, such as Temporal Difference Convolution or handcrafted Spatiotemporal Maps, by making simple adjustments to data pre-processing. Furthermore, optimizing the transformer configurations based on our GVT2RPM guidelines, such as introducing different biases via positional encodings and integrating spatiotemporal hierarchies via different scaling strategies, helps the model compete favorably with SOTA methods in both intra- and cross-dataset experiments. We also identified different behaviors in model performance between intra- and cross-dataset experiments, providing insights into selecting optimal configurations for different tasks. The proposed GVT2RPM guidelines were validated using three SOTA video transformers (MViTv2, UniFormer, and Video Swin) across five public datasets under intra- and cross-dataset settings. Our findings highlight that GVT2RPM is both generalizable to various transformer architectures and robust across diverse datasets.

CRediT authorship contribution statement

Hao Wang: Writing – review & editing, Writing – original draft, Project administration, Methodology, Formal analysis, Conceptualization. **Euijoon Ahn:** Writing – review & editing, Supervision, Conceptualization. **Andrew Joseph:** Writing – review & editing, Resources. **Faraz Pathan:** Writing – review & editing, Resources. **Kazuaki Negishi:** Writing – review & editing, Resources. **Jinman Kim:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Details of GVT2RPM-MViT-optimal configurations

We demonstrate the optimal configurations for adapting MViTv2 [55] to remote physiological measurement (RPM), i.e., GVT2RPM-MViT-optimal, in Table A.4.

Appendix B. Details of experiment results

We used rPPG-Toolbox [44] to evaluated five state-of-the-art (SOTA) methods, including DeepPhys [47], PhysNet [20], TS-CAN [53], EfficientPhys [26], and PhysFormer [36]. We compared them with GVT2RPM adapted version of MViTv2 [55], UniFormer [25], and Video Swin [24]. In addition to Mean Absolute Error (MAE), we also used metrics of Root Mean Square Error (RMSE) and Pearson Correlation Coefficient (ρ).

For intra-dataset experiments, we evaluated on four datasets, including MMPD-Simple [31], MMPD [31], UBFC-rPPG [33], and RLAP [32]. We show method performances of each fold in Tables B.5, B.6, B.7, B.8. The results are shown with *mean* $\pm$ *std*. Additionally, for three GVT2RPM-*-optimal methods, we visualized the differences between predictions and ground truth values by Bland-Altman plots in Figs B.7, B.8, B.9, B.10.

For cross-dataset experiments, we used five datasets, including MMPD-Simple [31], MMPD [31], UBFC-rPPG [33], RLAP [32], and UBFC-Phys [34]. We show detailed results in Tables B.9, B.10, B.11, B.12, B.13. The results are shown with *mean* $\pm$ *std*.

Appendix C. Dataset details

For cross-dataset experiments, the training set was spited into train/val set with ratio 8:2. For intra-dataset experiments, 3-fold cross-validation was used and we show the fold details in the following:

Table A.4
Optimal configurations for GVT2RPM-MViT.

Training set	Testing set	Frame format	Signal normalization	Positional encoding	Scaling strategy
MMPD-simple		DiffNorm	Yes	REL	Scale-2
MMPD		DiffNorm	No	REL	Scale-6
RLAP		DiffNorm	Yes	REL	Scale-2
UBFC-rPPG		DiffNorm	Yes	ABS	Scale-5
MMPD-simple	UBFC-rPPG	DiffNorm	No	CPE	Scale-0
	UBFC-Phys	DiffNorm	No	ABS	Scale-0
	RLAP	DiffNorm	No	CPE	Scale-2
MMPD	UBFC-rPPG	DiffNorm	No	CPE	Scale-0
	UBFC-Phys	DiffNorm	No	ABS	Scale-0
	RLAP	DiffNorm	No	ABS	Scale-0
RLAP	MMPD-simple	DiffNorm	Yes	ABS	Scale-0
	MMPD	DiffNorm	Yes	REL	Scale-2
	UBFC-rPPG	DiffNorm	Yes	CPE	Scale-3
	UBFC-Phys	DiffNorm	Yes	ABS	Scale-3
UBFC-rPPG	MMPD-simple	DiffNorm	Yes	REL	Scale-0
	MMPD	DiffNorm	Yes	REL	Scale-1
	UBFC-Phys	DiffNorm	No	ABS	Scale-3
	RLAP	DiffNorm	No	REL	Scale-3

Table B.5
Intra-dataset experiment results on UBFC-rPPG.

Methods	Fold 0			Fold 1			Fold 2		
	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑
DeepPhys	1.76 ± 1.22	4.08 ± 13.57	0.98 ± 0.07	0 ± 0	0 ± 0	1 ± 0	3.91 ± 1.50	5.96 ± 14.90	0.97 ± 0.09
TS-CAN	1.07 ± 0.91	2.94 ± 8.08	0.99 ± 0.05	0 ± 0	0 ± 0	1 ± 0	3.13 ± 1.50	5.48 ± 15.24	0.97 ± 0.08
EfficientPhys	1.07 ± 0.91	2.94 ± 8.08	0.99 ± 0.05	7.52 ± 6.30	20.35 ± 384.66	0.60 ± 0.30	1.86 ± 1.16	3.94 ± 9.75	0.99 ± 0.03
PhysNet	1.95 ± 1.21	4.12 ± 13.53	0.97 ± 0.07	1.86 ± 0.96	3.42 ± 6.75	0.98 ± 0.07	2.93 ± 1.43	5.19 ± 14.43	0.98 ± 0.08
PhysFormer	1.46 ± 0.98	3.28 ± 8.09	0.99 ± 0.06	1.27 ± 0.75	2.57 ± 5.10	0.99 ± 0.04	2.34 ± 1.13	4.12 ± 8.61	0.99 ± 0.05
GVT2RPM-MViT-optimal	1.56 ± 1.21	4.12 ± 13.52	0.99 ± 0.06	1.56 ± 0.97	3.31 ± 6.85	0.99 ± 0.06	0.87 ± 0.82	2.63 ± 6.55	0.99 ± 0.03
GVT2RPM-UniFormer-optimal	0.48 ± 0.46	1.46 ± 2.02	0.99 ± 0.02	1.85 ± 0.96	3.42 ± 6.75	0.98 ± 0.06	1.85 ± 1.15	3.94 ± 9.75	0.98 ± 0.07
GVT2RPM-Swin-optimal	1.36 ± 1.09	3.56 ± 11.61	0.99 ± 0.06	1.07 ± 0.75	2.50 ± 5.13	0.99 ± 0.03	2.14 ± 1.36	4.63 ± 14.40	0.99 ± 0.04

Table B.6
Intra-dataset experiment results on MMPD-simple.

Methods	Fold 0			Fold 1			Fold 2		
	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑
DeepPhys	17.50 ± 4.18	22.32 ± 159.07	−0.002 ± 0.33	2.72 ± 1.14	4.51 ± 12.20	0.60 ± 0.28	0.62 ± 0.39	1.39 ± 1.28	0.87 ± 0.17
TS-CAN	2.00 ± 0.76	3.21 ± 5.06	0.95 ± 0.10	1.23 ± 0.60	2.26 ± 2.99	0.92 ± 0.14	0.79 ± 0.50	1.78 ± 2.06	0.94 ± 0.12
EfficientPhys	1.54 ± 0.63	2.67 ± 3.53	0.97 ± 0.08	1.32 ± 0.62	2.51 ± 3.17	0.88 ± 0.15	0.88 ± 0.56	2.07 ± 2.79	0.91 ± 0.14
PhysNet	1.52 ± 0.77	2.96 ± 6.61	0.97 ± 0.08	1.32 ± 0.81	2.87 ± 6.00	0.85 ± 0.19	0.18 ± 0.11	0.39 ± 0.10	0.99 ± 0.04
PhysFormer	18.78 ± 3.03	21.30 ± 136.53	0.20 ± 0.33	3.60 ± 0.81	4.42 ± 6.04	0.51 ± 0.30	1.67 ± 0.74	2.87 ± 4.78	0.73 ± 0.24
GVT2RPM-MViT-optimal	0.51 ± 0.36	1.37 ± 1.53	0.99 ± 0.03	0.80 ± 0.38	1.79 ± 2.41	0.94 ± 0.10	0.64 ± 0.45	1.63 ± 2.39	0.85 ± 0.17
GVT2RPM-UniFormer-optimal	1.17 ± 0.57	2.27 ± 3.21	0.98 ± 0.06	1.19 ± 0.52	2.10 ± 2.20	0.90 ± 0.14	0.55 ± 0.37	1.37 ± 1.66	0.91 ± 0.13
GVT2RPM-Swin-optimal	1.02 ± 0.39	1.72 ± 1.54	0.99 ± 0.05	3.03 ± 1.09	4.72 ± 11.61	0.49 ± 0.28	1.91 ± 0.67	2.95 ± 3.86	0.56 ± 0.27

Table B.7
Intra-dataset experiment results on MMPD.

Methods	Fold 0			Fold 1			Fold 2		
	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑	MAE↓	RMSE↓	ρ ↑
DeepPhys	22.37 ± 1.56	28.85 ± 88.97	0.04 ± 0.09	20.30 ± 1.44	26.40 ± 78.66	0.05 ± 0.09	17.98 ± 1.23	23.02 ± 57.35	0.15 ± 0.09
TS-CAN	10.40 ± 1.36	19.01 ± 67.11	0.32 ± 0.08	10.95 ± 1.20	17.88 ± 50.47	0.48 ± 0.07	9.45 ± 1.06	15.55 ± 38.63	0.43 ± 0.07
EfficientPhys	13.67 ± 1.50	22.36 ± 76.37	0.19 ± 0.08	13.30 ± 1.35	20.75 ± 66.85	0.25 ± 0.08	11.96 ± 1.15	18.09 ± 44.92	0.37 ± 0.08
GVT2RPM-MViT-optimal	6.04 ± 0.86	11.72 ± 29.39	0.63 ± 0.06	8.31 ± 0.96	14.05 ± 35.28	0.58 ± 0.07	4.69 ± 0.66	9.08 ± 18.64	0.67 ± 0.06
GVT2RPM-UniFormer-optimal	6.91 ± 0.92	12.85 ± 31.05	0.52 ± 0.07	7.38 ± 0.96	13.54 ± 37.46	0.60 ± 0.06	5.81 ± 0.75	10.57 ± 22.04	0.54 ± 0.07
GVT2RPM-Swin-optimal	6.94 ± 1.04	13.97 ± 40.20	0.45 ± 0.07	8.34 ± 1.04	14.89 ± 41.41	0.58 ± 0.07	5.18 ± 0.76	10.33 ± 26.39	0.55 ± 0.07

C.1. MMPD-simple

Due to the difficulty of the original MMPD, authors created a subset to contain videos with stationary, skin tone type 3, and artificial light conditions.

For Fold 0, the training set is {'subject25', 'subject6', 'subject24', 'subject4', 'subject22', 'subject12', 'subject21', 'subject9', 'subject18', 'subject5', 'subject3', 'subject19', 'subject20'}, and the testing set is {'subject32', 'subject33', 'subject29', 'subject27'}.

For Fold 1, the training set is {'subject29', 'subject6', 'subject21', 'subject19', 'subject4', 'subject25', 'subject18', 'subject3', 'subject32', 'subject22', 'subject24', 'subject33', 'subject12'}, and the testing set is {'subject27', 'subject9', 'subject5', 'subject20'}.

For Fold 2, the training set is {'subject20', 'subject19', 'subject29', 'subject9', 'subject6', 'subject21', 'subject12', 'subject5', 'subject32',

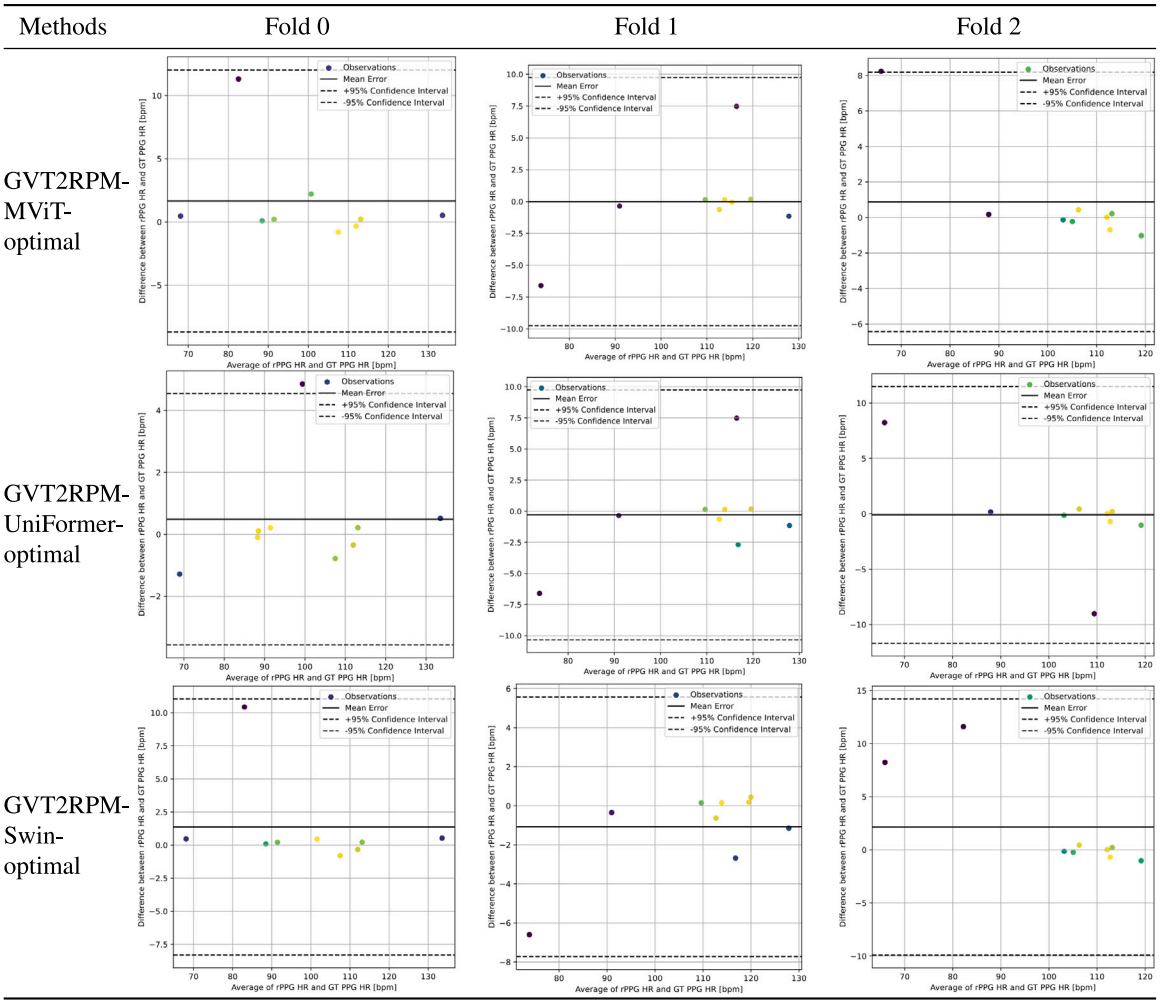


Fig. B.7. Intra-dataset experiment results on UBFC-rPPG.

Table B.8
Intra-dataset experiment results on RLAP.

Methods	Official split		
	MAE↓	RMSE↓	ρ ↑
DeepPhys	3.80 ± 0.69	7.90 ± 17.35	0.71 ± 0.07
TS-CAN	2.58 ± 0.57	6.33 ± 13.54	0.78 ± 0.06
EfficientPhys	2.98 ± 0.57	6.35 ± 11.71	0.80 ± 0.06
PhysNet	1.30 ± 0.32	3.38 ± 4.35	0.93 ± 0.04
PhysFormer	1.53 ± 0.37	3.98 ± 6.27	0.91 ± 0.04
GVT2RPM-MViT-optimal	1.32 ± 0.30	2.96 ± 3.01	0.95 ± 0.03
GVT2RPM-UniFormer-optimal	1.60 ± 0.38	3.73 ± 5.67	0.92 ± 0.04
GVT2RPM-Swin-optimal	1.64 ± 0.40	3.91 ± 5.62	0.91 ± 0.04

'subject25', 'subject3', 'subject33', 'subject18'}, and the testing set is {'subject27', 'subject4', 'subject22', 'subject24'}.

C.2. MMPD

For Fold 0, the training set is {'subject8', 'subject26', 'subject21', 'subject15', 'subject25', 'subject12', 'subject19', 'subject9', 'subject23', 'subject10', 'subject11', 'subject24', 'subject16', 'subject4', 'subject5', 'subject3', 'subject18', 'subject14', 'subject13', 'subject20', 'subject1', 'subject22', 'subject6', 'subject17', 'subject7', 'subject2'}, and the testing set is {'subject33', 'subject28', 'subject32', 'subject30', 'subject27', 'subject31', 'subject29'}.

Table B.9
Cross-dataset experiment results testing on UBFC-rPPG.

Method	Training set	Testing on UBFC-rPPG		
		MAE↓	RMSE↓	ρ ↑
DeepPhys	MMPD-simple	27.25 ± 3.84	36.90 ± 279.99	0.09 ± 0.16
	MMPD	29.72 ± 3.16	36.10 ± 195.37	0.21 ± 0.15
	RLAP	1.15 ± 0.40	2.87 ± 3.77	0.99 ± 0.02
TS-CAN	MMPD-simple	15.34 ± 3.55	27.63 ± 217.61	0.35 ± 0.15
	MMPD	16.22 ± 3.24	26.53 ± 181.05	0.47 ± 0.14
	RLAP	0.96 ± 0.37	2.60 ± 3.62	0.99 ± 0.02
EfficientPhys	MMPD-simple	15.11 ± 3.27	26.03 ± 196.11	0.36 ± 0.15
	MMPD	17.47 ± 3.41	28.15 ± 201.80	0.40 ± 0.14
	RLAP	1.95 ± 0.86	5.90 ± 26.31	0.95 ± 0.05
PhysNet	MMPD-simple	13.20 ± 2.66	21.69 ± 143.00	0.55 ± 0.13
	MMPD	N/A	N/A	N/A
	RLAP	8.20 ± 2.52	18.30 ± 140.44	0.51 ± 0.14
PhysFormer	MMPD-simple	19.11 ± 3.46	29.45 ± 211.58	0.14 ± 0.16
	MMPD	N/A	N/A	N/A
	RLAP	6.88 ± 2.06	15.03 ± 92.90	0.67 ± 0.12
GVT2RPM-MViT-optimal	MMPD-simple	1.46 ± 0.37	2.79 ± 2.43	0.99 ± 0.02
	MMPD	2.05 ± 0.61	4.48 ± 10.17	0.97 ± 0.04
	RLAP	1.21 ± 0.41	2.92 ± 3.76	0.99 ± 0.02

For Fold 1, the training set is {'subject20', 'subject7', 'subject11', 'subject4', 'subject25', 'subject32', 'subject6', 'subject15', 'subject33', 'subject16', 'subject22', 'subject26', 'subject23', 'subject19', 'subject10',

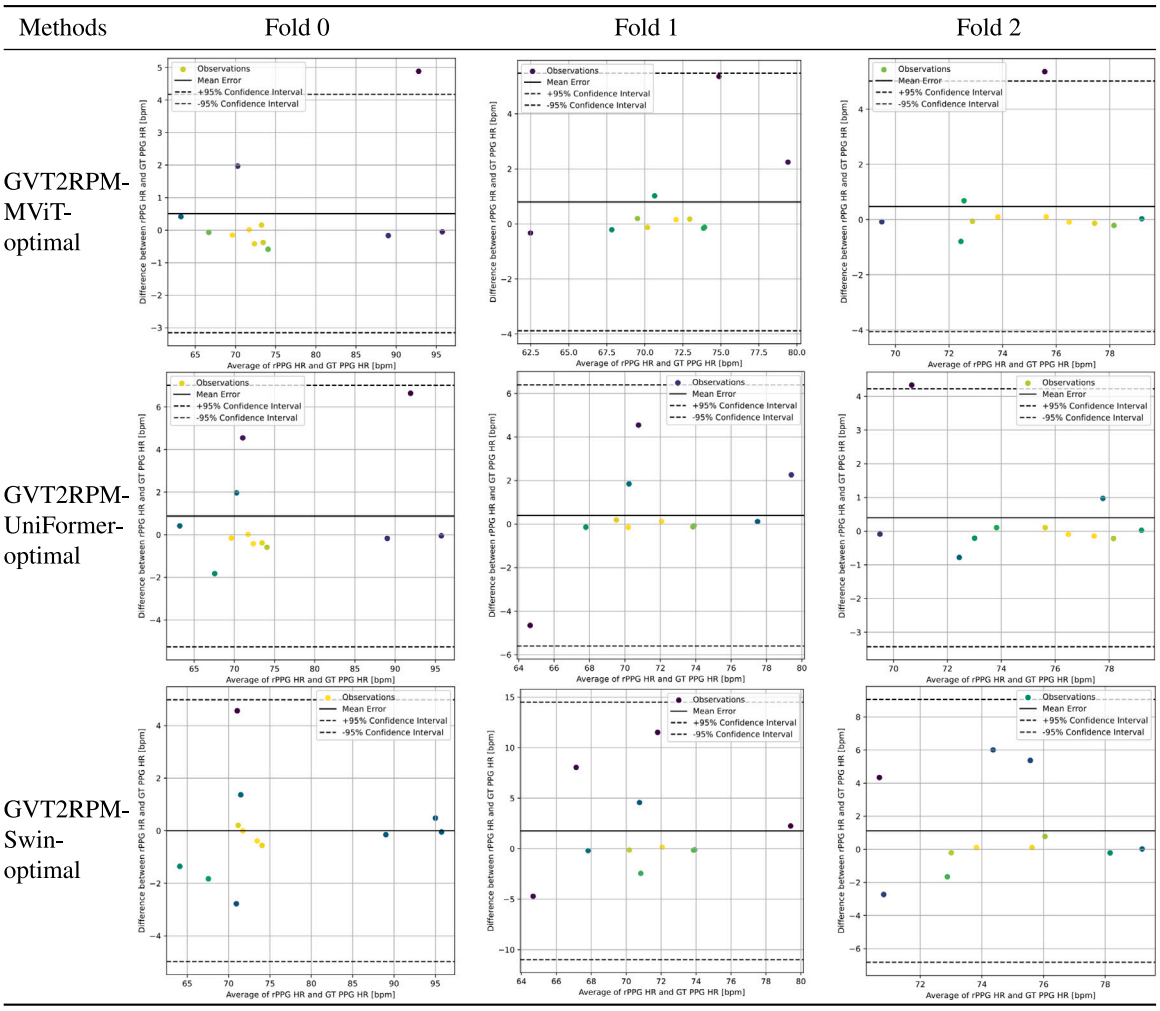


Fig. B.8. Intra-dataset experiment results on MMPD-simple.

'subject21', 'subject8', 'subject14', 'subject18', 'subject3', 'subject13', 'subject2', 'subject28', 'subject24', 'subject29', 'subject27'}, and the testing set is {'subject5', 'subject31', 'subject17', 'subject30', 'subject1', 'subject12', 'subject9'}.

For Fold 2, the training set is {'subject33', 'subject2', 'subject3', 'subject13', 'subject26', 'subject21', 'subject7', 'subject10', 'subject22', 'subject4', 'subject20', 'subject27', 'subject6', 'subject5', 'subject12', 'subject19', 'subject30', 'subject11', 'subject1', 'subject24', 'subject32', 'subject25', 'subject8', 'subject14', 'subject16', 'subject29'}, and the testing set is {'subject23', 'subject18', 'subject31', 'subject9', 'subject28', 'subject17', 'subject15'}.

C.3. RLAP

For Fold 0, the training set is {'subject6', 'subject46', 'subject53', 'subject48', 'subject58', 'subject31', 'subject13', 'subject52', 'subject56', 'subject14', 'subject36', 'subject18', 'subject38', 'subject9', 'subject49', 'subject50', 'subject43', 'subject4', 'subject44', 'subject23', 'subject15', 'subject57', 'subject33', 'subject11', 'subject24', 'subject55', 'subject27', 'subject40', 'subject45', 'subject2', 'subject3', 'subject42', 'subject35', 'subject20', 'subject21', 'subject32', 'subject10'}, the validation set is {'subject28', 'subject26', 'subject17', 'subject5', 'subject30'}, and the testing set is {'subject37', 'subject51', 'subject8', 'subject34', 'subject54', 'subject25', 'subject22', 'subject47', 'subject1', 'subject19', 'subject12', 'subject41', 'subject16', 'subject39'}.

C.4. UBFC-rPPG

For Fold 0, the training set is {'subject30', 'subject22', 'subject1', 'subject5', 'subject11', 'subject45', 'subject25', 'subject15', 'subject32', 'subject35', 'subject43', 'subject42', 'subject13', 'subject24', 'subject23', 'subject31', 'subject8', 'subject39', 'subject37', 'subject17', 'subject49', 'subject38', 'subject18', 'subject14', 'subject16', 'subject27', 'subject41', 'subject46', 'subject10', 'subject36', 'subject3', 'subject47', 'subject34'}, and the testing set is {subject26', 'subject4', 'subject33', 'subject9', 'subject40', 'subject12', 'subject44', 'subject48', 'subject20'}.

For Fold 1, the training set is {'subject45', 'subject8', 'subject12', 'subject20', 'subject27', 'subject38', 'subject30', 'subject13', 'subject23', 'subject47', 'subject16', 'subject5', 'subject41', 'subject26', 'subject25', 'subject35', 'subject22', 'subject31', 'subject10', 'subject49', 'subject44', 'subject3', 'subject18', 'subject17', 'subject46', 'subject9', 'subject39', 'subject32', 'subject42', 'subject43', 'subject37', 'subject24', 'subject40'}, and the testing set is {'subject4', 'subject1', 'subject36', 'subject15', 'subject34', 'subject33', 'subject11', 'subject14', 'subject48'}.

For Fold 2, the training set is {'subject46', 'subject49', 'subject12', 'subject13', 'subject35', 'subject9', 'subject5', 'subject17', 'subject18', 'subject3', 'subject26', 'subject20', 'subject4', 'subject31', 'subject14', 'subject24', 'subject11', 'subject16', 'subject40', 'subject45', 'subject1', 'subject32', 'subject34', 'subject41', 'subject33', 'subject36', 'subject43', 'subject42', 'subject39', 'subject10', 'subject48', 'subject38', 'subject37'}, and the testing set is {'subject47', 'subject25', 'subject44', 'subject27', 'subject8', 'subject22', 'subject30', 'subject23', 'subject15'}.

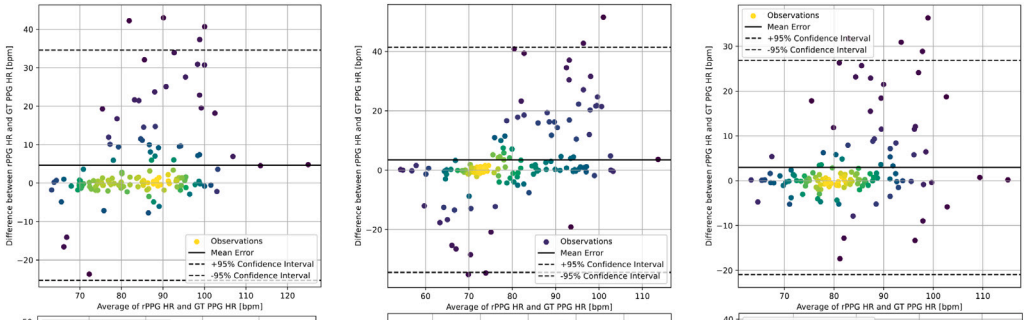
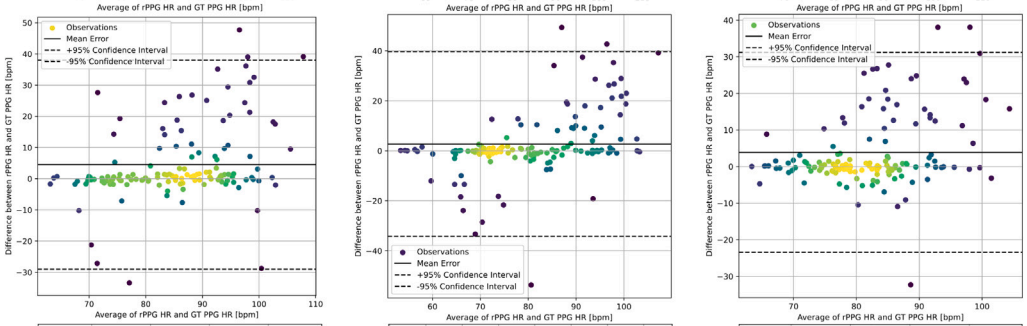
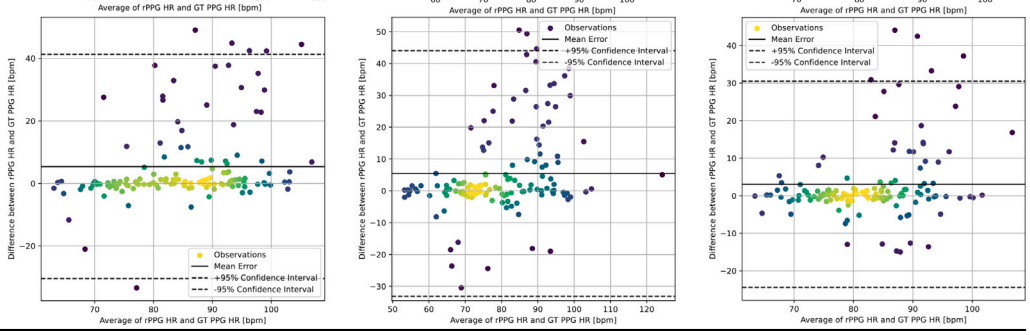
Methods	Fold 0	Fold 1	Fold 2
GVT2RPM-MViT-optimal			
GVT2RPM-UniFormer-optimal			
GVT2RPM-Swin-optimal			

Fig. B.9. Intra-dataset experiment results on MMPD.

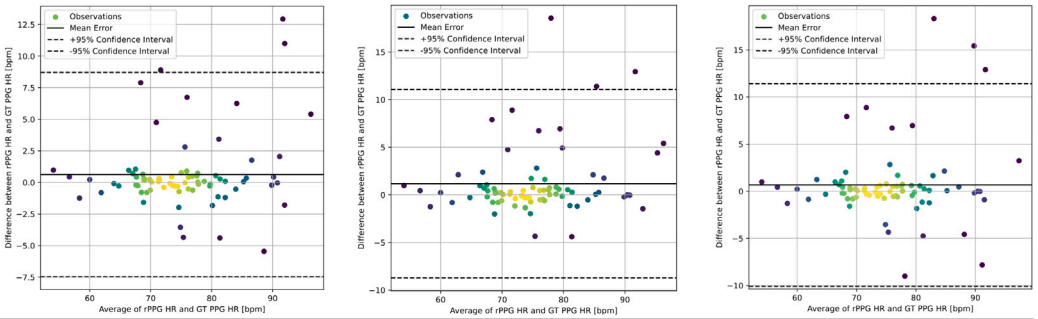
Dataset	GVT2RPM-MViT-optimal	GVT2RPM-UniFormer-optimal	GVT2RPM-Swin-optimal
RLAP			

Fig. B.10. Intra-dataset experiment results on RLAP.

Table B.10
Cross-dataset experiment results testing on UBFC-Phys.

Method	Training set	Testing on UBFC-Phys		
		MAE↓	RMSE↓	ρ ↑
DeepPhys	UBFC-rPPG	6.01 ± 0.88	10.70 ± 28.05	0.64 ± 0.08
	MMPD-simple	13.36 ± 1.26	18.43 ± 52.96	0.20 ± 0.10
	MMPD	14.65 ± 1.28	19.49 ± 56.77	0.07 ± 0.10
	RLAP	4.81 ± 0.68	8.32 ± 16.15	0.76 ± 0.07
TS-CAN	UBFC-rPPG	5.27 ± 0.68	8.63 ± 14.86	0.74 ± 0.07
	MMPD-simple	7.91 ± 1.02	12.91 ± 34.83	0.49 ± 0.09
	MMPD	7.19 ± 0.95	11.96 ± 30.15	0.53 ± 0.08
	RLAP	4.36 ± 0.65	7.89 ± 16.25	0.78 ± 0.06
EfficientPhys	UBFC-rPPG	6.07 ± 0.86	10.57 ± 24.47	0.64 ± 0.08
	MMPD-simple	5.14 ± 0.79	9.49 ± 27.11	0.70 ± 0.07
	MMPD	5.79 ± 0.80	9.95 ± 21.92	0.67 ± 0.07
	RLAP	4.28 ± 0.67	7.95 ± 16.55	0.78 ± 0.06
PhysNet	UBFC-rPPG	4.54 ± 0.75	8.80 ± 24.62	0.75 ± 0.07
	MMPD-simple	7.28 ± 0.87	11.42 ± 25.85	0.50 ± 0.09
	MMPD	N/A	N/A	N/A
	RLAP	4.48 ± 0.74	8.66 ± 25.33	0.76 ± 0.07
PhysFormer	UBFC-rPPG	5.13 ± 0.73	8.98 ± 18.49	0.73 ± 0.07
	MMPD-simple	9.23 ± 0.95	13.28 ± 30.64	0.37 ± 0.09
	MMPD	N/A	N/A	N/A
	RLAP	4.48 ± 0.70	8.36 ± 22.12	0.76 ± 0.07
GVT2RPM-MViT-optimal	UBFC-rPPG	4.22 ± 0.64	7.65 ± 13.59	0.79 ± 0.06
	MMPD-simple	5.09 ± 0.60	7.97 ± 11.99	0.76 ± 0.07
	MMPD	4.32 ± 0.64	7.76 ± 14.83	0.78 ± 0.06
	RLAP	4.17 ± 0.71	8.26 ± 23.49	0.77 ± 0.06

Table B.11
Cross-dataset experiment results testing on MMPD-simple.

Method	Training set	Testing on MMPD-simple		
		MAE↓	RMSE↓	ρ ↑
DeepPhys	UBFC-rPPG	2.98 ± 0.81	6.35 ± 21.14	0.82 ± 0.09
	RLAP	1.87 ± 0.61	4.60 ± 14.38	0.88 ± 0.07
TS-CAN	UBFC-rPPG	1.61 ± 0.40	3.22 ± 4.20	0.94 ± 0.05
	RLAP	1.32 ± 0.37	2.87 ± 3.58	0.96 ± 0.04
EfficientPhys	UBFC-rPPG	0.91 ± 0.25	2.01 ± 1.60	0.98 ± 0.03
	RLAP	0.97 ± 0.25	2.02 ± 1.43	0.98 ± 0.03
PhysNet	UBFC-rPPG	2.69 ± 0.91	6.95 ± 31.79	0.70 ± 0.10
	RLAP	1.52 ± 0.42	3.23 ± 4.32	0.95 ± 0.05
PhysFormer	UBFC-rPPG	7.38 ± 1.97	15.53 ± 106.46	0.14 ± 0.15
	RLAP	2.55 ± 0.78	5.96 ± 19.24	0.78 ± 0.09
GVT2RPM-MViT-optimal	UBFC-rPPG	1.87 ± 0.82	6.07 ± 31.55	0.78 ± 0.09
	RLAP	0.79 ± 0.22	1.70 ± 1.15	0.98 ± 0.03

Table B.12
Cross-dataset experiment results testing on MMPD.

Method	Training set	Testing on MMPD		
		MAE↓	RMSE↓	ρ ↑
DeepPhys	UBFC-rPPG	17.72 ± 0.67	24.63 ± 37.43	0.14 ± 0.04
	RLAP	16.74 ± 0.72	24.82 ± 40.87	0.05 ± 0.04
TS-CAN	UBFC-rPPG	13.52 ± 0.62	20.84 ± 31.27	0.22 ± 0.04
	RLAP	13.34 ± 0.63	20.97 ± 32.46	0.21 ± 0.04
EfficientPhys	UBFC-rPPG	13.08 ± 0.64	20.99 ± 32.99	0.20 ± 0.04
	RLAP	12.69 ± 0.62	20.38 ± 31.73	0.21 ± 0.04
PhysNet	UBFC-rPPG	9.94 ± 0.48	15.84 ± 20.38	0.32 ± 0.04
	RLAP	9.15 ± 0.50	15.67 ± 21.65	0.35 ± 0.04
PhysFormer	UBFC-rPPG	12.98 ± 0.54	19.01 ± 27.16	0.13 ± 0.04
	RLAP	9.99 ± 0.49	15.91 ± 21.19	0.32 ± 0.04
GVT2RPM-MViT-optimal	UBFC-rPPG	10.23 ± 0.48	15.94 ± 20.38	0.31 ± 0.04
	RLAP	8.28 ± 0.44	13.90 ± 16.63	0.45 ± 0.03

Table B.13
Cross-dataset experiment results testing on RLAP.

Method	Training set	Testing on RLAP		
		MAE↓	RMSE↓	ρ ↑
DeepPhys	UBFC-rPPG	4.90 ± 0.44	10.25 ± 15.77	0.54 ± 0.04
	MMPD-simple	10.65 ± 0.56	15.64 ± 21.27	0.17 ± 0.05
	MMPD	10.89 ± 0.54	15.50 ± 19.66	0.19 ± 0.05
TS-CAN	UBFC-rPPG	3.20 ± 0.31	7.07 ± 9.37	0.76 ± 0.03
	MMPD-simple	5.89 ± 0.43	10.55 ± 13.08	0.51 ± 0.04
	MMPD	7.07 ± 0.47	11.89 ± 18.62	0.37 ± 0.06
EfficientPhys	UBFC-rPPG	3.77 ± 0.38	8.49 ± 12.95	0.66 ± 0.04
	MMPD-simple	3.89 ± 0.35	8.06 ± 10.65	0.67 ± 0.04
	MMPD	4.05 ± 0.38	8.34 ± 13.54	0.64 ± 0.04
PhysNet	UBFC-rPPG	3.39 ± 0.37	8.43 ± 10.89	0.68 ± 0.04
	MMPD-simple	6.11 ± 0.42	10.36 ± 12.71	0.49 ± 0.04
	MMPD	N/A	N/A	N/A
PhysFormer	UBFC-rPPG	4.44 ± 0.46	10.18 ± 17.35	0.54 ± 0.04
	MMPD-simple	9.12 ± 0.64	15.73 ± 27.08	0.32 ± 0.05
	MMPD	N/A	N/A	N/A
GVT2RPM-MViT-optimal	UBFC-rPPG	2.83 ± 0.38	7.39 ± 13.73	0.72 ± 0.04
	MMPD-simple	3.02 ± 0.34	6.80 ± 11.10	0.76 ± 0.04
	MMPD	2.77 ± 0.35	6.78 ± 10.82	0.79 ± 0.03

Data availability

The authors do not have permission to share data.

References

[1] Telehealth is here to stay, Nature Med. 27 (7) (2021) 1121, http://dx.doi.org/10.1038/s41591-021-01447-x.

[2] B. Huang, S. Hu, Z. Liu, C.-L. Lin, J. Su, C. Zhao, L. Wang, W. Wang, Challenges and prospects of visual contactless physiological monitoring in clinical study, Npj Digit. Med. 6 (1) (2023) 231, http://dx.doi.org/10.1038/s41746-023-00973-x.

[3] B.P. Yan, W.H.S. Lai, C.K.Y. Chan, A.C.K. Au, B. Freedman, Y.C. Poh, M.-Z. Poh, High-throughput, contact-free detection of atrial fibrillation from video with deep learning, JAMA Cardiol. 5 (1) (2020) 105–107, http://dx.doi.org/10.1001/jamacardio.2019.4004.

[4] W. Verkruysse, L.O. Svaasand, J.S. Nelson, Remote plethysmographic imaging using ambient light, Opt. Express 16 (26) (2008) 21434–21445, http://dx.doi.org/10.1364/OE.16.021434.

[5] J. Allen, Photoplethysmography and its application in clinical physiological measurement, Physiol. Meas. 28 (3) (2007) R1–R39, http://dx.doi.org/10.1088/0967-3334/28/3/R01.

[6] M. Elgendi, R. Fletcher, Y. Liang, N. Howard, N.H. Lovell, D. Abbott, K. Lim, R. Ward, The use of photoplethysmography for assessing hypertension, Npj Digit. Med. 2 (1) (2019) 60, http://dx.doi.org/10.1038/s41746-019-0136-7.

[7] T. Pereira, N. Tran, K. Gadhoumi, M.M. Pelter, D.H. Do, R.J. Lee, R. Colorado, K. Meisel, X. Hu, Photoplethysmography based atrial fibrillation detection: a review, Npj Digit. Med. 3 (1) (2020) 3, http://dx.doi.org/10.1038/s41746-019-0207-9.

[8] X. Li, J. Chen, G. Zhao, M. Pietikäinen, Remote heart rate measurement from face videos under realistic situations, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4264–4271, http://dx.doi.org/10.1109/CVPR.2014.543.

[9] M.-Z. Poh, D.J. McDuff, R.W. Picard, Advancements in noncontact, multiparameter physiological measurements using a webcam, IEEE Trans. Biomed. Eng. 58 (1) (2011) 7–11, http://dx.doi.org/10.1109/TBME.2010.2086456.

[10] D. Wedekind, A. Trumpp, F. Gaetjen, S. Rasche, K. Matschke, H. Malberg, S. Zaunseder, Assessment of blind source separation techniques for video-based cardiac pulse extraction, J. Biomed. Opt. 22 (3) (2017) 035002, http://dx.doi.org/10.1117/1.JBO.22.3.035002.

[11] W. Wang, A.C. den Brinker, S. Stuijk, G. de Haan, Algorithmic principles of remote PPG, IEEE Trans. Biomed. Eng. 64 (7) (2017) 1479–1491, http://dx.doi.org/10.1109/TBME.2016.2609282.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, vol. 25, Curran Associates, Inc, 2012, URL: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, http://dx.doi.org/10.48550/arxiv.1409.1556, arXiv.Org.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Las Vegas, NV, USA, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90, URL: http://ieeexplore.ieee.org/document/7780459/.

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks, IEEE, 2015, pp. 4489–4497, http://dx.doi.org/10.1109/ICCV.2015.510.

[16] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A Closer Look at Spatiotemporal Convolutions for Action Recognition, IEEE, 2018, pp. 6450–6459, http://dx.doi.org/10.1109/CVPR.2018.00675.

[17] C. Feichtenhofer, H. Fan, J. Malik, K. He, SlowFast Networks for Video Recognition, IEEE, 2019, pp. 6201–6210, http://dx.doi.org/10.1109/ICCV.2019.00630.

[18] Y. Qiu, Y. Liu, J. Arteaga-Falconi, H. Dong, A. El Saddik, EVM-CNN: Real-time contactless heart rate estimation from facial video, IEEE Trans. Multimed. 21 (7) (2018) 1778–1787.

[19] R. Spetlik, V. Franc, J. Cech, J. Matas, Visual heart rate estimation with convolutional neural network, 2018.

[20] Z. Yu, X. Li, G. Zhao, Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks, 2019.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Attention is All you Need, vol. 30, Curran Associates, Inc, 2017, http://dx.doi.org/10.48550/arXiv.1706.03762, URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021, URL: https://openreview.net/forum?id=YicbFdNTTy.

[23] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale Vision Transformers, IEEE, 2021, pp. 6804–6815, http://dx.doi.org/10.1109/ICCV48922.2021.00675.

[24] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, 2022, pp. 3202–3211.

[25] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, Y. Qiao, UniFormer: Unifying convolution and self-attention for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 45 (10) (2023) 12581–12600, http://dx.doi.org/10.1109/TPAMI.2023.3282631.

[26] X. Liu, B. Hill, Z. Jiang, S. Patel, D. McDuff, EfficientPhys: Enabling simple, fast and accurate camera-based cardiac measurement, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2023, pp. 4997–5006, http://dx.doi.org/10.1109/WACV56688.2023.00498, URL: http://doi.ieeecomputersociety.org/10.1109/WACV56688.2023.00498.

[27] X. Zhang, W. Sun, H. Lu, Y. Chen, Y. Ge, X. Huang, J. Yuan, Y. Chen, Self-similarity prior distillation for unsupervised remote physiological measurement, IEEE Trans. Multimed. (2024).

[28] Z. Yu, Y. Shen, J. Shi, H. Zhao, P.H. Torr, G. Zhao, PhysFormer: Facial video-based physiological measurement with temporal difference transformer, 2022, pp. 4186–4196.

[29] H. Wang, E. Ahn, J. Kim, Self-supervised representation learning framework for remote physiological measurement using spatiotemporal augmentation loss, Proc. AAAI Conf. Artif. Intell. 36 (2) (2022) 2431–2439, http://dx.doi.org/10.1609/aaai.v36i2.20143.

[30] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan, M. Shah, Transformers in vision: A survey, ACM Comput. Surv. 54 (10s) (2022) http://dx.doi.org/10.1145/3505244.

[31] J. Tang, K. Chen, Y. Wang, Y. Shi, S. Patel, D. McDuff, X. Liu, MMPD: Multi-domain mobile video physiology dataset, in: 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC, 2023, pp. 1–5, http://dx.doi.org/10.1109/EMBC40787.2023.10340857.

[32] K. Wang, Y. Wei, M. Tong, J. Gao, Y. Tian, Y. Ma, Z. Zhao, PhysBench: A benchmark framework for rPPG with a new dataset and baseline, 2023, http://dx.doi.org/10.48550/arxiv.2305.04161, arXiv.Org.

[33] S. Bobbia, R. Macwan, Y. Benezeth, A. Mansouri, J. Dubois, Unsupervised skin tissue segmentation for remote photoplethysmography, Pattern Recognit. Lett. 124 (supl) (2017) 82–90, http://dx.doi.org/10.1016/j.patrec.2017.10.017.

[34] R. Meziati Sabour, Y. Benezeth, P. De Oliveira, J. Chappe, F. Yang, UBFC-phys: A multimodal database for psychophysiological studies of social stress, IEEE Trans. Affect. Comput. 14 (1) (2023) 1, http://dx.doi.org/10.1109/TAFFC.2021.3056960.

[35] X. Niu, S. Shan, H. Han, X. Chen, Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation, IEEE Trans. Image Process. 29 (2019) 2409–2423, http://dx.doi.org/10.1109/TIP.2019.2947204, publisher: IEEE.

[36] Y. Li, C. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, C. Feichtenhofer, MViTv2: Improved multiscale vision transformers for classification and detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 4794–4804, http://dx.doi.org/10.1109/CVPR52688.2022.00476, URL: http://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00476.

[37] G. Bertasius, H. Wang, L. Torresani, Is space-time attention all you need for video understanding? 2021, p. 4, 2, issue: 3.

[38] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, C. Schmid, Vivit: A video vision transformer, 2021, pp. 6836–6846.

[39] D. Neimark, O. Bar, M. Zohar, D. Asselmann, Video transformer network, 2021, pp. 3163–3172.

[40] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 9992–10002, http://dx.doi.org/10.1109/ICCV48922.2021.00986, URL: http://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00986.

[41] X. Liu, Y. Zhang, Z. Yu, H. Lu, H. Yue, J. Yang, rPPG-MAE: Self-supervised pretraining with masked autoencoders for remote physiological measurements, IEEE Trans. Multimed. (2024).

[42] Z. Yu, Y. Shen, J. Shi, H. Zhao, Y. Cui, J. Zhang, P. Torr, G. Zhao, PhysFormer++: Facial video-based physiological measurement with SlowFast temporal difference transformer, Int. J. Comput. Vis. 131 (6) (2023) 1307–1330, http://dx.doi.org/10.1007/s11263-023-01758-1.

[43] Z. Yu, B. Zhou, J. Wan, P. Wang, H. Chen, X. Liu, S.Z. Li, G. Zhao, Searching multi-rate and multi-modal temporal enhanced networks for gesture recognition, IEEE Trans. Image Process. 30 (2021) 5626–5640, http://dx.doi.org/10.1109/TIP.2021.3087348.

[44] X. Liu, G. Narayanswamy, A. Paruchuri, X. Zhang, J. Tang, Y. Zhang, Y. Wang, S. Sengupta, S. Patel, D. McDuff, rPPG-Toolbox: Deep Remote PPG Toolbox, NIPS '23, Curran Associates Inc, 2023, http://dx.doi.org/10.48550/ARXIV.2210.00716, URL: https://arxiv.org/abs/2210.00716.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), PyTorch: An Imperative Style, High-Performance Deep Learning Library, Curran Associates, Inc, 2019, pp. 8024–8035, http://dx.doi.org/10.48550/arXiv.1912.01703, URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[46] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019, http://dx.doi.org/10.48550/arXiv.1711.05101, URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[47] W. Chen, D. McDuff, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks, Springer International Publishing, Cham, 2018, pp. 356–373.

[48] W. Wang, S. Stuijk, G. de Haan, Exploiting spatial redundancy of image sensor for motion robust rPPG, IEEE Trans. Biomed. Eng. 62 (2) (2015) 415–425, http://dx.doi.org/10.1109/TBME.2014.2356291, 216 citations (Semantic Scholar/DOI) [2023-11-07].

[49] S. Ioffe, C. Szegedy, in: F. Bach, D. Blei (Eds.), Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, 2015, pp. 448–456, URL: https://proceedings.mlr.press/v37/ioffe15.html.

[50] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[51] M.A. Islam, S. Jia, N.D.B. Bruce, How much position information do convolutional neural networks encode? 2020, URL: https://openreview.net/forum?id=rJeB36NKvB.

[52] X. Chu, Z. Tian, B. Zhang, X. Wang, C. Shen, Conditional positional encodings for vision transformers, 2023, URL: https://openreview.net/forum?id=3KWnuT-R1bh.

[53] X. Liu, J. Fromm, S. Patel, D. McDuff, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Multi-Task Temporal Shift Attention Networks for On-Device Contactless Vitals Measurement, vol. 33, Curran Associates, Inc, 2020, pp. 19400–19411, URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e1228be46de6a0234ac22ded31417bc7-Paper.pdf.

[54] F. Isensee, P.F. Jaeger, S.A.A. Kohl, J. Petersen, K.H. Maier-Hein, nnU-net: a self-configuring method for deep learning-based biomedical image segmentation, Nature Methods 18 (2) (2021) 203–211, http://dx.doi.org/10.1038/s41592-020-01008-z.

[55] J. Lin, C. Gan, S. Han, TSM: Temporal shift module for efficient video understanding, 2019, pp. 7082–7092, http://dx.doi.org/10.1109/ICCV.2019.00718, URL: http://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00718.