Phase Unwrapping via Fully Exploiting Global and Local Spatial Dependencies

Yuhui Quan[a], Xin Yao[a], Zhifeng Chen[b,*], Hui Ji[c]

[a]*School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510000, Guangdong, China*
[b]*School of Physics and Materials Science, Guangzhou University, Guangzhou, Guangdong 510006, Guangzhou, 510000, Guangdong, China*
[c]*Department of Mathematics, National University of Singapore, 119076, Singapore*

Abstract

Phase unwrapping (PU) is the process of extracting the authentic phase image from its noisy wrapped measurements, playing a crucial role in scientific imaging techniques. PU requires solving a challenging non-linear ill-posed problem. particularly in the presence of noticeable noise. In recent years, deep learning has emerged as a promising approach for PU. Inspired by the success of convolutional neural networks (CNNs) in image restoration, many existing works trained CNNs for PU. However, due to the locality of convolutional kernels, CNNs are not efficient in capturing global spatial dependencies, a critical cue for PU. As an alternate, recent studies employed recurrent neural networks (RNNs) defined on handcrafted pixel paths. Nonetheless, a limited number of pre-defined pixel paths cannot fully exploit global spatial

*This work was supported in part by National Natural Science Foundation of China under Grant 62072188, in part by Natural Science Foundation of Guangdong Province under Grants 2022A1515011755 and 2023A1515012841, and in part by Singapore MOE AcRF under Grant A-8000981-00-00.

*Corresponding author.

Email addresses: csyhquan@scut.edu.cn (Yuhui Quan), xinyao240@gmail.com (Xin Yao), chenzf@gzhu.edu.cn (Zhifeng Chen), matjh@nus.edu.sg (Hui Ji)

dependencies existing in complex phase structures. In this paper, we introduce a vision transformer (ViT) model that effectively captures both global and local spatial dependencies using a hierarchical structure with a multi-scale process. The proposed ViT model employs a series of global transformer blocks to capture global spatial dependencies at the roughest scale. The resulting global features are used to guide a set of local transformer blocks to analyze local spatial dependencies in a coarse-to-fine progressive manner for unwrapping. Extensive experiments show that, our proposed ViT model produces higher-quality unwrapped phases over existing CNN/RNN-based methods, while maintaining a lightweight nature.
Keywords: Phase Unwrapping; Transformer Models; Deep Learning; Phase Imaging

1. Introduction

Phase Unwrapping (PU) is a fundamental problem in image sensing, whose goal is to retrieve authentic phases from the wrapped ones; see Fig. 1 for an illustration. Due to their inherent operational characteristics, many image sensing systems produce wrapped phase measurements, typically constrained in $[-\pi, \pi)$. For example, quantitative phase imaging techniques such as phase-contrast microscopy and digital holography [1; 2], magnetic resonance imaging through quantitative susceptibility mapping [3], 3D scanning using fringe projection profilometry (FPP) [4; 5; 6; 7; 8; 9], Doppler radar imaging [10], and interferometric synthetic aperture radar imaging [11; 12], among others. Particularly, PU plays an critical role in FPP, a widely used optical technique for 3D shape measurement. Various FPP methods dif-

fer in how they acquire the wrapped phase, which results in different formation of wrapped phase. Thus, Different FPP methods employ distinct approaches to project, capture, and process fringe patterns. Representative methods include Fourier Transform Profilometry (FTP), Phase-Shifting Profilometry (PSP), MoireProfilometry (MP), Computer-Generated MoireProfilometry (CGMP), Modulation Measuring Profilometry (MMP), and Phase-Differencing Profilometry (PDP).

In FTP [13; 14], a single fringe pattern is projected onto the object, and the deformed fringe pattern is captured. The wrapped phase is then obtained by applying a Fourier transform to the captured image, filtering out unwanted frequencies, and performing an inverse Fourier transform. The wrapped phase is then extracted from the resulting complex image. PSP [15] involves projecting a series of fringe patterns with known phase shifts (typically three or more) onto the object. Then, the wrapped phase is calculated from pixel intensity variations across these patterns. MP [16] generates the wrapped phase by analyzing interference patterns (Moire fringes) formed by superimposting grating on two sets of gratings (one on the object and one as a reference), where the modulating phase of the moire pattern caused by the object's surface geometry encodes the wrapped phase. An extension of MP, CGMP [17; 18] allows precise digitial control of the reference grating. MMP [19; 20] focuses on analyzing the modulation changes in the projected fringe patterns due to surface topography, typically combined with phase-shifting techniques for improved accuracy. Finally, PDP [21] employs number-theoretical Temporal PU method [22] to compute phase-shifting deformed patterns, that are both computational efficient and robust, particu-

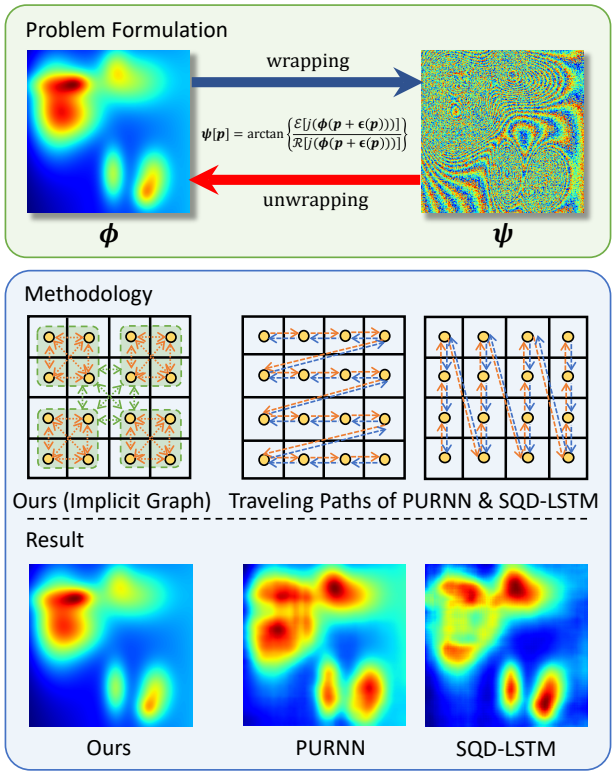

Figure 1: PU task and methodology/results of existing RNN-based methods: PURNN [23] and SQD-LSTM [24], and our approach. our approach models hierarchical interactions via blending global connections with localized interactions.

larly in high-speed 3D measurement. In all these FPP methods, resolving the ambiguities in wrapped phases via PU is essential for accurately capturing and processing images, which are critical for the precise reconstruction of 3D shapes.

Although PU is of great importance, it remains a challenging task. Challenges such as measurement noise, inherent system inconsistencies, and abrupt phase variations, can introduce noticeable errors during unwrapping, leading to inaccurate phase restoration. Let $\boldsymbol{\phi}(\boldsymbol{p}) \in \mathbb{R}$ denote the true phase on the

vector coordinate $\boldsymbol{p}$ and $\boldsymbol{\psi}(\boldsymbol{p}) \in [-\pi, \pi)$ be its noisy wrapped version. They are related by

$$\boldsymbol{\psi}[\boldsymbol{p}] = w(\boldsymbol{\phi}(\boldsymbol{p}) + \boldsymbol{\epsilon}(\boldsymbol{p}))), \tag{1}$$

where $\boldsymbol{\epsilon}$ denotes measurement noise. The operator w denotes the wrapping operator which maps any phase $\theta \in \mathbb{R}$ to the range $[-\pi, \pi)$:

$$w(\theta) = [(\theta + \pi) \bmod 2\pi] - \pi,$$

where $[\cdot \mod 2\pi]$ denotes the modulo operation with a modulus of 2π. Another often-used formulation of phase wrapping is based on the wrap count, which is expressed as:

$$\boldsymbol{\psi}[\boldsymbol{p}] = \boldsymbol{\phi}(\boldsymbol{p}) + 2\pi \cdot \boldsymbol{k}(\boldsymbol{p}) - pi + \bar{\boldsymbol{\epsilon}}(\boldsymbol{p}), \tag{2}$$

where $\bar{\boldsymbol{\epsilon}}$ denotes noise depending on both truth phase θ and measurement noise $\boldsymbol{\epsilon}$. The operator $\boldsymbol{k} = \text{round}(\frac{\boldsymbol{\phi}-\boldsymbol{\psi}}{2\pi}) \in \mathbb{Z}$ denotes the map of wrap counts, indicating the number of times a phase value has been wrapped around by 2π.

Clearly,, the solution to (2) is non-unique, underscoring ill-posedness of the PU problem. Furthermore, PU is highly sensitive to noise. Any naive approach, such as direct integration of wrapped phases can lead to erroneous results, primarily stemming from the accumulation of noise through the integration path.

PU can be categorized into spatial (SPU) and temporal (TPU) methods. In the past, various SPU approaches were developed, including path-following, filtering, and optimization techniques. Similarly, numerous TPU methods were proposed, such as gray code, phase coding, phase shifting, and

fringe amplitude encoding algorithms. Each approach has its strengths and limitations, particularly regarding noise handling and path selection.

Recently, inspired by the potent modeling capability of Deep Neural Networks (DNNs), an increasing number of studies have utilized DNNs for PU [25], showing advantages in terms of accuracy and efficiency over traditional handcrafted methods. There are mainly two methodologies in existing studies on DNN-based PU: regression-based methods (*e.g.* [26; 24; 27; 28]) that train a DNN to directly predict unwrapped phases from the wrapped input, using a loss function measuring pixel-wise errors between predicted and authentic phases; and classification-based methods (*e.g.* [29; 30; 31; 32]) that trains a DNN to predict wrap counts via a pixel-wise classification loss function that interprets wrap counts as class labels, thereby turning PU into a segmentation-like problem.

The majority of existing DNN-based PU methods employ convolutional neural networks (CNNs); see *e.g.* [26; 33; 29; 30; 28; 27; 2; 31; 3; 2; 6; 34]. While CNNs excel at extracting local spatial features due to their convolutional layers with localized receptive fields, they are not efficient for modeling global dependencies, due to the linear growth of a CNN's receptive field with added layers. This diminishes CNNs' suitability to PU for which a holistic understanding of the entire image is crucial. Indeed, the accurate unwrapping on a given pixel needs to understand not just its immediate neighborhood, but also how that region relates to distant parts of the image, as the phase values in one region can be influenced by the phase jumps/wraps occurring in far-off regions. As a result, global dependencies are critical for PU, particularly when handling complex phase structures.

Emerging as an alternative to better grasping global dependencies, recurrent neural networks (RNNs) are leveraged by [23; 24] to model distant regional information. Yet, their performance has often been constrained to the manually specified traveling paths within feature spaces (see Fig. 7). This manual path selection introduces significant simplification, given the complex and variable characteristics of spatial dependencies of phase images. Simply put, a limited number of predefined paths cannot adequately encompass the intricate spatial relationships. Meanwhile, incorporating an extensive array of traveling paths for comprehensive coverage becomes computationally prohibitive.

An ideal DNN architecture for PU must efficiently capture and leverage both local and global spatial dependencies within a phase image. Toward this end, we explore the exploitation of the strong capability of transformers [35] in capturing global spatial dependencies. Unlike classification which concerns global contexts and semantics, most classic image restoration tasks, such as denoising and super-resolution, care more about local structures and details. Together with the high computational cost of employing a global attention mechanism, existing Vision Transformers (ViT) employed in these tasks typically restrict their attention to local regions and rarely employ position encoding. However, PU differs much from these tasks. While it recognizes the importance of local dependencies, PU also necessitates a global attention mechanism to tap into global dependencies for resolving ambiguities. Hence, a transformer tailored for PU is needed to efficiently exploit global dependencies without an excessive computational cost.

In response to the demands on both computational efficiency and inte-

gration of global-and-local dependencies, we introduce a ViT model called PUTFormer (PU TransFormer), employing a multi-scale strategy that seamlessly integrates joint global and local analysis while maintaining computational efficiency. Our PUTFormer processes tokens across various scales, derived from a multi-scale patch embedding mechanism. On the coarser scales, global dependencies are discerned using global transformer blocks, which are used subsequently to guide local transformer blocks in predicting features of unwrapped phases in a coarse-to-fine progression. In essence, it implicitly establishes a hierarchical spatial relationship graph (see Fig. 1), blending global connections with localized spatial interactions. This obviates the necessity for the manual path definitions commonly found in the RNN-based methodologies.

Furthermore, recognizing the pivotal role of spatial order in PU, we integrate positional encoding into PUTFormer. This imparts a natural understanding of spatial orientation, distinguishing it from the ViT models used in many other image reconstruction tasks where positional encoding is often omitted. Following the same practice of [29; 24], the PUTFormer is trained using a gradient-domain loss that bypasses the instability issue caused by the equivalent class of ground-truth (GT) phases. Extensive experiments under different settings have demonstrated the advantages of PUTFormer over existing works, in terms of both unwrapping accuracy (see Fig. 1) and computational efficiency.

To conclude, there are three contributions in this paper:

- Introducing the first ViT model tailored for PU;
- Proposing a specialized hierarchical design to optimize the performance

and efficiency of ViT for PU;

- Developing a lightweight DNN for PU that achieves state-of-the-art results.

The rest of this paper is organized as follows. Section 2 performs a literature review. Section 3 presents the details of our proposed PUTFormer. Section 4 is devoted to experimental evaluation. Finally, Section 5 draws a conclusion.

2. Related Work

2.1. Non-deep-learning Methods for PU

PU can be roughly categorized into spatial (SPU) and temporal (TPU). Early studies tackled SPU primarily using three conventional methods: path-following, filtering, and optimization. Path-following-based methods (*e.g.* [36; 37; 11; 38; 39; 40; 41]) perform PU by integrating along chosen paths. For example, diamond (rthombus) type strategy [42] employs a diamond stencil is often employed to evaluate phase differences between neighboring pixels, and curtain-type strategy [36] unwraps the phase in a sweeping manner, either horizontally or vertically across the image. However, both the imperfectness of path selection and the noise can result in amplified errors during path integration. Aiming for better noise robustness, filtering-based methods (*e.g.* [43; 44]) adapt non-linear denoising filters for concurrent PU and denoising. In a different vein, optimization-based methods (*e.g.* [45; 46; 47; 48; 49]) recast SPU as an optimization problem regularized by some handcrafted image priors for improving noise robustness and guiding path selection. Nevertheless, optimization-based methods are likely to diminish the dynamic range

of phase values [50], and the used handcrafted priors may be over-simplistic for complex phase structures.

There are also extensive studies on TPU. Gray code algorithms [51; 52], use sequential binary patterns to uniquely encode phase shifts for robust unwrapping. Phase coding algorithms [53; 54] enhance accuracy by encoding phase with stair-like patterns to reduce ambiguity. Phase shifting algorithms [7; 55] use multiple sinusoidal fringe patterns to compute phase shifts for high-resolution 3D reconstruction. Fringe amplitude encoding algorithms [56] extract absolute phase information by encoding fringe intensity modulations, improving robustness against noise and discontinuities.

2.2. DNN-based Methods for PU

2.2.1. Regression-based methods

Regression-based DNNs are tailored for the end-to-end prediction of unwrapped phases. Dardikman *et al.* [26] used residual CNNs. Wang *et al.* [28] and Qin *et al.* [27] used U-shaped CNNs. Peng *et al.* [6] used both residual and U-shaped CNNs. To exploit global spatial dependencies, these methods necessitate stacking many layers for a sufficiently large receptive field. To make this efficient, Zhang *et al.* [32] inserted an edge-enhanced self-attention (SA) into the bottleneck of a U-shaped CNN.

To better exploit global dependencies, Ryu *et al.* [23] constructed an RNN defined across pixels. Further, Perera *et al.* [24] introduced an RNN enhanced by Long Short-Term Memories (LSTM) [57]. These methods need to pre-define several paths for RNN construction. Given computational constraints, this limited number of paths cannot fully harness the global spatial dependencies inherent in a phase image. In contrast, the PUTFormer is a

transformer with a specialized structure, optimizing spatial dependency extraction while maintaining computational efficiency.

2.2.2. Classification-based methods

Rather than directly predict unwrapped phases, classification-based methods such as [3; 2] train a DNN to predict wrap counts. This strategy recasts PU as a segmentation task, treating neighboring pixels with identical wrap counts as segments. Consequently, established segmentation DNNs can be employed for PU. However, this strategy may be susceptible to noise. Thus, Zhang *et al.* [30] inserted a denoiser before wrap count prediction. Alternatively, Zhang *et al.* [31] introduced a refinement module to post-process the errors caused by noise. Similarly, Spoorthi *et al.* [33; 29] applied Gaussian filtering to the phases unwrapped by a densely-connected CNN. In addition to the necessity of dedicated denoising modules, classification-based methods can grapple with the extensive array of resultant classes for phases with wide-range values.

2.2.3. Blending DNNs with conventional techniques

Luo *et al.* [58] employed a classification DNN specifically to remove invalid data points, enhancing the robustness of PU. Jiang *et al.* [50] merged a semantic segmentation DNN with path-following and non-linear filtering techniques. Instead of learning an end-to-end mapping, Yang *et al.* [1] utilized an untrained CNN to re-parameterize the latent phase image and optimized it to match the measurements. This method is costly as it trains individual CNNs for different samples.

2.3. ViT Models for Image Restoration

Building on their success in image classification, ViT models have recently been proposed for a variety of image restoration and reconstruction tasks; see *e.g.* [59; 60; 61; 62; 63; 64; 65; 66]. These tasks often deal with high-resolution images. Applying ViT models then can be computationally intensive due to the global attention computation. Moreover, as these tasks focus on understanding the local structures and details rather than the global content, most existing ViT models used in these tasks restricted their attention mechanisms in local windows. Different from those classic image restoration tasks, PU necessitates a global attention mechanism to exploit global dependencies, as a critical step toward resolving ambiguities in PU. Our proposed PUTFormer effectively bridges this gap without imposing high computational costs.

3. Methodology

As illustrated in Fig. 2, the PUTFormer is architecturally crafted to predict the true unwrapped phase image from an input wrapped phase image. Central to its design is the integration of multi-resolution analysis, allowing for comprehensive phase insights from different scales, while optimizing computational efficiency. Specifically, we have

1) **Multi-Scale Patch Embedding**. For multi-resolution analysis, the multi-scale patch embedding module is proposed for generating tokens that represent the input at multiple scales. Following this token generation, each token undergoes Positional Encoding (PE), ensuring that spatial relationships within the input data are preserved and understood by subsequent transformer blocks.

2) **Global analysis using Global Transformer Blocks (GTBs)**. For understanding global dependencies, GTBs are employed for discerning and capturing long-range dependencies across the input image, deployed at the coarsest scale for computational efficiency.

3) **Progressive Local Analysis with Local Transformer Blocks (LTBs)**. Guided by the global understanding from the GTBs, the PUTFormer architecture transitions to a coarse-to-fine unwrapping process, executed by the LFBs in a progressive manner. The guidance from global features ensure the stability of the unwrapping process.

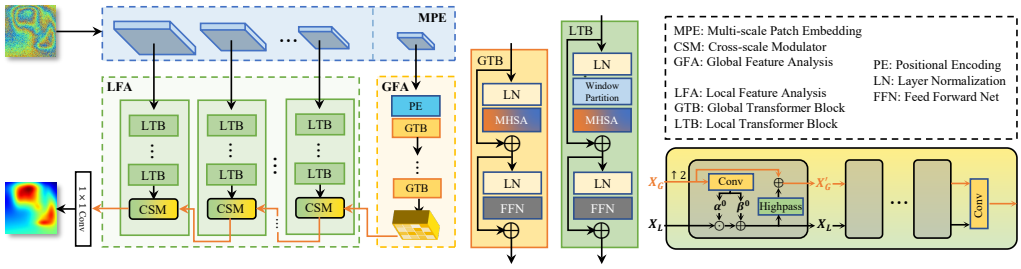

Figure 2: Diagram for illustrating the architecture of PUTFormer.

3.1. Multi-Scale Patch Embedding with PE

The essence of the multi-scale patch embedding lies in its capability to tokenize image patches across varying scales within the feature space. This is achieved using a succession of 3×3 convolutional layers, each employing a stride of 2 for downsampling. With every convolutional layer, tokens corresponding to patches at differing scales are generated. The features derived from deeper layers represent coarser scales. For each spatial location, token extraction is streamlined by taking the feature vector along the channel

dimension as a token.

Incorporating positional information into extracted tokens is very helpful, especially for PU where spatial dependencies significantly impact outcomes. This is achieved through PE. Given an input tensor $\boldsymbol{Z} \in \mathbb{R}^{H\times W\times D}$, the PE layer, denoted by $\mathrm{PE} : \mathbb{R}^{H\times W\times D} \to \mathbb{R}^{H\times W\times D}$, is formulated as follows:

$$\boldsymbol{P}[x, y, d] = \sin(x/\omega^{2d/D}) + \cos(y/\omega^{2d/D}), \tag{3}$$

$$\mathrm{PE}(\boldsymbol{Z}) = \boldsymbol{Z} + \boldsymbol{P}, \tag{4}$$

where $\omega = 10^4$ in our implementation. Here (x, y) denotes the row and column indices of the 2D grid, and d is the index along the encoding dimension.

3.2. Exploiting Global Dependencies with GTBs

GTBs serve a critical role in PUTFormer by processing tokens at the coarsest scale. Their objective is twofold: (i) to interpret the overall spatial layout of a phase image; and (ii) to discern relations between regions with varying wrap counts. A GTB, denoted by T_{G}, is a traditional transformer block comprised of a layer normalization (LayerNorm) [67], a multi-head self-attention (MHSA), and a feed-forward network (FFN), which can be expressed as:

$$\boldsymbol{X}' = \boldsymbol{X} + \mathrm{MHSA}(\mathrm{LayerNorm}(\boldsymbol{X})), \tag{5}$$

$$T_{\mathrm{G}}(\boldsymbol{X}) = \boldsymbol{X}' + \mathrm{FFN}(\mathrm{LayerNorm}(\boldsymbol{X}')). \tag{6}$$

Within these operations, LayerNorm normalizes the feature by computing their mean and variance, then scales and shifts them using learnable parameters. This aids in stabilizing and accelerating training.

For a set of tokens stored as $\boldsymbol{Z} = [\boldsymbol{z}_1; \cdots; \boldsymbol{z}_L] \in \mathbb{R}^{L \times D}$, the MHSA seeks to derive new token representations by assessing interdependence among every pair of input tokens. For the h-th of H attention head, all tokens undergo a linear transform resulting in $\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h \in \mathbb{R}^{L \times D}$, which represent queries, keys and values respectively:

$$(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = (\boldsymbol{Z}\boldsymbol{W}_h^{\mathrm{Q}}, \boldsymbol{Z}\boldsymbol{W}_h^{\mathrm{K}}, \boldsymbol{Z}\boldsymbol{W}_h^{\mathrm{V}}), \tag{7}$$

where $\boldsymbol{W}_h^{\mathrm{Q}}, \boldsymbol{W}_h^{\mathrm{K}}, \boldsymbol{W}_h^{\mathrm{V}}$ are learnable matrices. Subsequently, attention weights are derived, determining the extent to which each token interacts with its counterparts. This is achieved through the calculation of similarity scores between queries and keys, leading to:

$$\mathrm{Head}_h = \mathrm{softmax}(\boldsymbol{Q}_h \boldsymbol{K}_h^T / \sqrt{D}) \boldsymbol{V}_h. \tag{8}$$

To consolidate results from all attention heads, the MHSA output is given by

$$\mathrm{MHSA}(\boldsymbol{Z}) = \mathrm{concat}([\mathrm{Head}_1, \mathrm{Head}_2, \cdots, \mathrm{Head}_H]) \boldsymbol{W}^O, \tag{9}$$

where $\boldsymbol{W}^O$ is a learnable matrix dedicated to fusing the results of the different attention heads.

To infuse the model with increased non-linearity, we utilize an FFN structure used in [60]:

$$\mathrm{FFN}(\boldsymbol{X}) = (\mathrm{GELU}(\boldsymbol{X}\boldsymbol{W} \odot \boldsymbol{X}\boldsymbol{W}_2)\boldsymbol{X}\boldsymbol{W}_3, \tag{10}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$ are learnable matrices, $\mathrm{GELU}(\cdot)$ is the Gaussian Error Linear Unit [68], and $\odot$ denotes element-wise multiplication.

Overall, the computational flow of the global transformer stage with N global transformer blocks can be expressed as:

$$\boldsymbol{X}_G = T_G^{(N)} \circ T_G^{(N-1)} \circ \cdots T_G^{(1)}(\mathrm{PE}(\boldsymbol{X})), \tag{11}$$

where $T_G^{(n)}$ denotes the n-th global transformer block.

3.3. Retrieving Local Dependencies via LTBs

High-resolution embedding tokens are hypothesized to convey essential low-level information, enhancing the network's ability to retrieve intricate local details and dependencies. Similar to [69] [70], for high-resolution features of dimension $H \times W \times C$, we adopt a partitioning strategy by reshaping these features into tensors of dimensions $H/p \times W/p \times p^2 \times C$. Within each window, spatial inter-dependencies are then retrieved through p^2 feature points. Our empirical observations show that this granular modeling approach improves the PU process, leading to the retrieval of more local details.

It's worth noting that the design of LTB mirrors its global counterpart in many respects. However, a distinguishing feature lies in the application of the SA mechanism, which is executed within each window. A LTB, denoted by $T_L(\boldsymbol{X})$, can be expressed as

$$\boldsymbol{X}' = \boldsymbol{X} + \mathrm{MHSA}(\mathrm{Norm}(\mathrm{Partition}(\boldsymbol{X}))), \tag{12}$$

$$T_L(\boldsymbol{X}) = \boldsymbol{X}' + \mathrm{FFN}(\boldsymbol{X}'), \tag{13}$$

where $\mathrm{Partition}(\cdot)$ is the aforementioned partition operation.

3.4. Cross-Scale Modulator

To materialise the benefit from the guidance of global dependencies captured in GTBs, it needs to appropriately handle the features from different

scales. We identify two feature types: the global feature $\boldsymbol{X}_G$ and the local feature $\boldsymbol{X}_L$. The difference between two feature types is more than just spatial granularity. That is, $\boldsymbol{X}_G$ represents a coarser, globally-consistent unwrapping, emphasizing overall phase consistency, whereas $\boldsymbol{X}_L$ embodies finer local details, though wrapped and potentially cluttered with noise. The effective interplay between these features can improve the unwrapping accuracy by having the best of both worlds: global consistency and local precision.

Toward this end, we introduce a Cross-scale Modulator (CSM) to integrate these two types of features. Note that the coarser unwrapped $\boldsymbol{X}_G$ typically exhibits a richer spectrum of phase values in contrast to the tightly wrapped nature of $\boldsymbol{X}_L$. Leveraging this disparity, we first modulate $\boldsymbol{X}_L$'s phase spectrum to align more closely with that of $\boldsymbol{X}_G$:

$$\boldsymbol{\alpha} = f_{\boldsymbol{\theta}_1}(\boldsymbol{X}_G), \quad \boldsymbol{\beta} = f_{\boldsymbol{\theta}_2}(\boldsymbol{X}_G), \tag{14}$$

$$\boldsymbol{X}_L := \boldsymbol{\alpha} \odot \boldsymbol{X}_L + \boldsymbol{\beta}. \tag{15}$$

Subsequently, we mine $\boldsymbol{X}_L$ for local details, extracted via high-pass filtering operations (implemented through residual blocks), and infuse them into $\boldsymbol{X}_G$. By this way, the CSM iteratively adjusts and refines local details.

3.5. Loss Function

Consider a true/wrapped phase image pair $(\boldsymbol{\phi}, \boldsymbol{\psi})$. Let $\mathcal{F}_{\boldsymbol{\theta}}$ denote our PUTFormer parameterized by $\boldsymbol{\theta}$. For any $\boldsymbol{\psi}$, note that $\boldsymbol{\phi} + 2\pi c$ $(\forall c \in \mathbb{Z})$ retains the structural consistency of $\boldsymbol{\phi}$ and results in the same wrapped image, $\boldsymbol{\phi}$. Hence, the solution to PU corresponds to an equivalence class:

$$\Phi = \{\boldsymbol{\phi} + 2\pi c : \forall c \in \mathbb{Z}\}. \tag{16}$$

As a result, directly training $\mathcal{F}_{\boldsymbol{\theta}}$ to predict $\boldsymbol{\phi}$ challenges this inherent nature, potentially leading to issues with training stability and generalization. For instance, given two training pairs $(\boldsymbol{\phi}, \boldsymbol{\psi})$ and $(\boldsymbol{\phi} + 2\pi, \boldsymbol{\psi})$, the NN trained to approximate both $\boldsymbol{\phi}$ and $\boldsymbol{\phi} + 2\pi$ could make the convergence difficult and may result in an undesired average prediction.

Sharing a similar spirit with [29; 24], we avoid such issues by employing a training loss that quantifies prediction errors within the gradient domain:

$$\mathcal{L}(\boldsymbol{\theta}) := \|\nabla_x \boldsymbol{\phi} - \nabla_x \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\psi})\|_2^2 + \|\nabla_y \boldsymbol{\phi} - \nabla_y \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\psi})\|_2^2, \tag{17}$$

where ∇_x and ∇_y denote the gradient operators along the x and y axes, respectively. In comparison to [29; 24], we use a purely gradient-domain loss. In essence, our approach leverages only the relative values of the true phase image for guidance. Due to the invariance of the gradient operators to a constant pixel value shift, *i.e.*, $\nabla(\boldsymbol{\phi} + c) = \nabla\boldsymbol{\phi}$, we can effectively resolve those issues linked to the solution set Φ.

4. Experiments

4.1. Data, Protocol, and Implementation Details

In supervised learning, the quality of the training data plays a critical role in determining the model's performance. Following the approach in [24], we synthesize unwrapped phase images by generating a mixture of randomly distributed Gaussian blobs with varying parameters. The phase wrapping operation is then simulated as described in Eq. 1. In practice, unwrapped phase patterns exhibit significant variation, making it challenging to create a training dataset that accounts for all possible phase patterns. To ensure an

accurate evaluation of the generalization performance of the proposed and compared methods, all methods are tested across diverse testing datasets that differ from the training data. These include real-world datasets such as RME [28] and InSAR, in addition to the synthetic Gaussian mixture model.

For a better evaluation, we make two modifications on the scheme of [24]. First, we increase the diversity and complexity of phase patterns, by enlarging the maximal Gaussian cluster number P from 4 to 16. Second, we widen the range of phase values from $[-7 \cdot 2\pi, 7 \cdot 2\pi]$ to $[-10 \cdot 2\pi, 10 \cdot 2\pi]$, increasing the challenge. Totally 5000 paired samples are generated for training, with SNRs uniformly sampled from $\{0, 5, 10, 20, 60\}$. The resolution of all wrapped phase images is 256×256.

A set of DNN-based PU methods is chosen for performance comparison: PURNN [23], PhaseNet2.0 [29], SQD-LSTM [24], and EESANet [32]. Additionally, we include three representative transformer DNNs for general image restoration tasks, including SwinIR [71], Uformer [61] and Restormer [60], as references. For these three transformer models as well as SQD-LSTM, we retrain them using their official codes. The other three methods do not have public codes. We faithfully implement them according to the instructions in the literature, reproducing their results. Following [24], we use Normalized Root Mean Square Error (NRMSE) as the accuracy metric, which is defined as

$$\mathrm{NRMSE}_{\boldsymbol{\phi}',\boldsymbol{\phi}} = \frac{\|\boldsymbol{\phi}' - \boldsymbol{\phi}\|_2}{\sqrt{WH}\big(\max(\boldsymbol{\phi}) - \min(\boldsymbol{\phi})\big)}, \tag{18}$$

where $\boldsymbol{\phi}' \in \mathbb{R}^{H \times W}$ is the predicted unwrapped phase image. Before calculation, min-max normalization is applied to $\boldsymbol{\phi}'$ so that $\boldsymbol{\phi}'$ has the same range of values as $\boldsymbol{\phi}$. For convenience, all NRMSE values are reported in the units

of 10^{-2}.

We train the PUTFormer using the Adam optimizer [72] with an initial learning rate of 1×10^{-3}. The learning rate is halved every 5×10^4 iterations and the total iteration number is 3×10^5. The whole training process takes nearly 10 hours on an Nvidia GTX 1080Ti GPU. Our PyTorch code will be released on GitHub upon paper's acceptance.

4.2. Performance Evaluation

To have a comprehensive evaluation from different perspectives, we construct seven test cases detailed as follows.

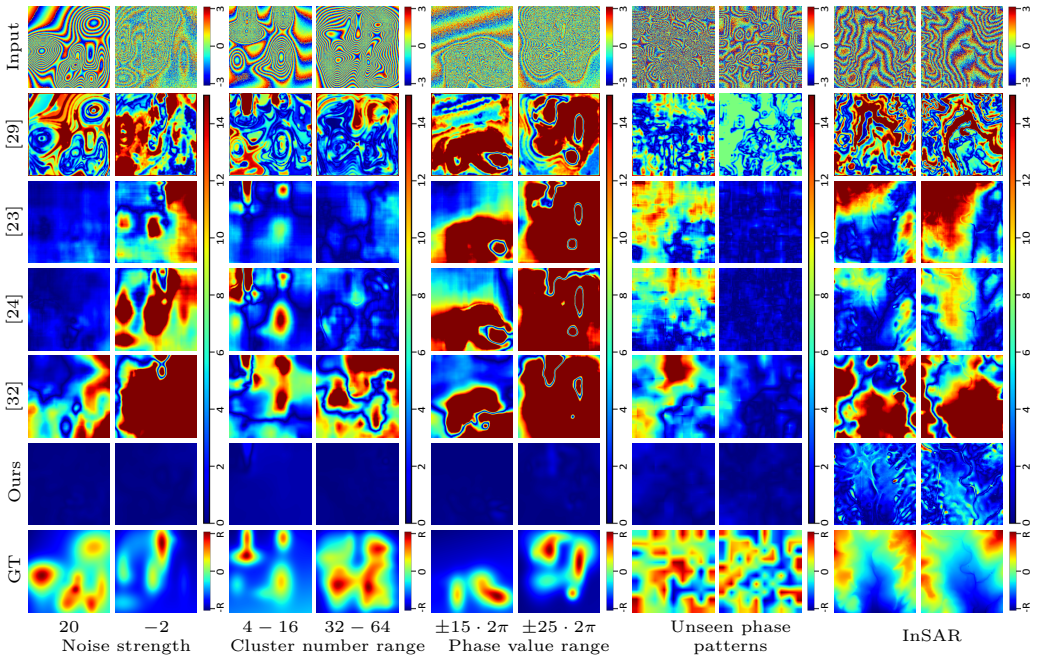

Figure 3: Visual inspection of residual images under different test settings.

4.2.1. Robustness test against various noise strengths

Test data is constructed following Sec. 4.1 but with a wider range of noise strengths, additionally including a nosier case with SNR=-2 and a noise-free case with SNR=∞ (*i.e.* $\boldsymbol{n} = \mathbf{0}$). We generate 1000 test samples per noise strength.

See Table 1 for the results. PUTFormer consistently performs the best across all noise settings, including the unseen SNRs of 2 and ∞. The average NRMSE of PUTFormer is around 1/3 of SQD-LSTM, the top competitor among the compared PU-dedicated DNNs, and also noticeably better than Restormer, a general transformer-based model. Additionally, PUTFormer shows smaller performance decrease when handling noisier data, compared to other methods. See Fig. 3 for visual inspection of the results for a sample case. For easier inspection, the residual images are shown. Apparently, PUTFormer is better in recovering both global structures and local patterns, attributed to its capability of exploiting both global and local spatial dependencies for PU. In comparison, SQD-LSTM only exploits spatial dependencies along a few pre-defined regular paths, thus less effective and generalizable. In addition, our method shows good robustness to large noise, e.g., SNR=-2, as shown in Fig. 3.

4.2.2. Generalization test against higher complexities

We construct a test set following Sec. 4.1 and change the range of the Gaussian glob number to four cases, respectively: 1$\sim$4, 4$\sim$16, 16$\sim$32 and 32$\sim$64. The last two settings form phase structures with higher complexity than the training data. We generate 1000 test samples per range.

As shown in Table 2, PUTFormer ranks the first among all the competi-

Table 1: NRMSE results under various SNRs.

Method	∞	60	20	10	5	0	-2	Average
PURNN [23]	2.45	2.40	2.47	2.44	2.59	3.86	6.11	3.19
PhaseNet2.0 [29]	9.90	9.86	9.78	9.74	9.98	9.79	10.08	9.88
SQD-LSTM [24]	0.88	0.82	0.86	0.85	0.86	1.27	3.27	1.26
EESANet [32]	9.78	10.34	10.04	10.05	10.93	11.81	15.96	11.27
SwinIR [71]	7.56	7.41	7.42	7.67	7.80	8.85	11.55	8.32
Uformer [61]	0.96	0.96	0.97	0.97	0.94	1.16	1.87	1.12
Restormer [60]	0.84	0.85	0.87	0.86	0.80	0.87	1.56	0.95
PUTFormer	**0.13**	**0.13**	**0.13**	**0.14**	**0.16**	**0.36**	**1.19**	**0.32**

tors, with noticeably superior performance. Overall, its average NRMSE is around 1/5 of SQD-LSTM (top-performer of compared PU-dedicated DNNs). When more Gaussian globs are introduced, the performance drop of PUTFormer is less than SQD-LSTM. The reason is probably that, spatial dependencies become much richer as the phase complexity increases, which cannot be fully captured by SQD-LSTM that uses a limited number of fixed regular paths. In contrast, PUTFormer is path-free and exploits spatial dependencies with a pair-wise manner, thereby more generalizable to complex phase structures. See also Fig. 3 for a visual inspection. Additionally, even compared to the general Restormer model, PUTFormer still achieves better results, demonstrating the effectiveness of its architecture.

4.2.3. Generalization test on unseen phase ranges

In this experiment, the test samples are generated as described in Sec. 4.1, but with extended phase ranges of $[-15 \cdot 2\pi, 15 \cdot 2\pi]$, $[-20 \cdot 2\pi, 20 \cdot 2\pi]$, and $[-25 \cdot 2\pi, 25 \cdot 2\pi]$, respectively. For each range, 1,000 test samples are

Table 2: NRMSE results under various ranges of cluster numbers.

Method	1-4	4-16	16-32	32-64	Average
PURNN [23]	1.34	1.57	2.17	2.77	1.96
PhaseNet2.0 [29]	9.48	9.60	9.83	10.16	9.82
SQD-LSTM [24]	<u>0.83</u>	<u>0.92</u>	1.04	1.10	1.00
EESANet [32]	10.73	11.46	10.78	10.49	10.87
SwinIR [71]	6.03	6.03	6.47	7.12	6.41
Uformer [61]	0.86	0.99	1.27	1.47	1.15
Restormer [60]	0.93	0.97	<u>0.95</u>	<u>0.98</u>	<u>0.96</u>
PUTFormer	**0.16**	**0.19**	**0.23**	**0.24**	**0.21**

generated.

The quantitative results in Table 3 indicate that PUTFormer generalizes effectively across a wide range of phase values and consistently outperforms the compared methods, including Restormer. The improvement is especially notable compared to other PU-dedicated methods. The visual comparison for an example case is shown in Fig. 3, where the output from PUTFormer is visibly closer to the GT compared to the baselines. For example, when the phase value range becomes much larger than the original one, *e.g.*, $[-20 \cdot 2\pi, 20 \cdot 2\pi]$ and $[-25 \cdot 2\pi, 25 \cdot 2\pi]$, SQD-LSTM fails to recover the phases, while the PUTFormer still works well.

The performance gain of PUTFormer largely come from its ability to effectively utilize global consistency when unwrapping the phase, which is crucial as the phase values deviate further from the training range. Furthermore, this global consistency is well balanced with local features through modulating schemes, enabling a robust recovery aross different phase ranges.

Table 3: NRMSE results under various phase ranges.

Method	$\pm 10 \cdot 2\pi$	$\pm 15 \cdot 2\pi$	$\pm 20 \cdot 2\pi$	$\pm 25 \cdot 2\pi$	Average
PURNN [23]	1.55	3.85	7.03	11.25	5.92
PhaseNet2.0 [29]	10.04	10.47	10.29	11.16	10.49
SQD-LSTM [24]	0.94	1.63	3.37	6.08	3.01
EESANet [32]	11.40	11.16	12.24	13.47	12.07
SwinIR [71]	6.08	6.46	6.92	8.34	6.95
Uformer [61]	0.98	1.49	2.67	4.53	2.42
Restormer [60]	<u>0.85</u>	<u>1.02</u>	<u>1.63</u>	<u>2.78</u>	<u>1.57</u>
PUTFormer	**0.24**	**0.34**	**0.89**	**1.82**	**0.81**

4.2.4. Generalization test on unseen phase patterns

We use a different scheme, the one employed in [28], to generate test samples with unseen phase patterns. This scheme creates a GT phase image by interpolating and rescaling a small random matrix sampled from $\mathcal{U}(\mathbf{0}, \mathbf{1})$. The wrapping process follows Sec. 4.1, with 1000 samples generated per SNR.

As observed in Table 4, directly evaluating pre-trained models on the unseen patterns results in performance drop for all methods. Yet, our PUTFormer still performs better than other PU-dedicated DNNs. Compared to Restormer, it performs better on 3/5 noise strengths. We also retrain all models using 5000 samples generated with the scheme of [28]. The results are also listed in Table 4. We can see that the results of all models become better after retraining. In this case, PUTFormer is top-1 on 4/5 noise strengths. See also Fig. 3 for a qualitative comparison.

Table 4: NRMSE of pre-/re-trained models on unseen patterns.

	Method	60	20	10	5	0	Average
Pre-trained	PURNN [23]	12.79	12.70	12.86	13.22	15.27	13.37
	PhaseNet2.0 [29]	12.86	12.90	12.79	12.93	14.42	13.18
	SQD-LSTM [24]	9.64	9.63	9.62	10.04	12.97	10.38
	EESANet [32]	25.92	25.61	25.19	25.33	27.76	25.96
	SwinIR [71]	9.45	9.65	9.52	9.83	11.39	9.97
	Uformer [61]	6.48	6.50	6.53	6.71	9.49	7.14
	Restormer [60]	5.93	5.94	6.35	**6.43**	**9.12**	**6.75**
	PUTFormer	**5.90**	**5.91**	**6.20**	6.47	9.33	6.76
Re-trained	PURNN [23]	2.05	2.05	2.14	2.30	4.31	2.57
	PhaseNet2.0 [29]	9.99	9.77	9.74	9.94	10.29	9.95
	SQD-LSTM [24]	2.03	2.01	2.01	2.06	3.58	2.34
	EESANet [32]	4.42	4.32	4.18	4.27	5.18	4.47
	SwinIR [71]	4.73	4.74	4.70	4.77	5.11	4.81
	Uformer [61]	1.33	1.30	1.38	1.60	1.96	1.51
	Restormer [60]	0.82	0.80	0.81	0.80	0.93	0.83
	PUTFormer	**0.51**	**0.50**	**0.52**	**0.68**	**1.75**	**0.79**

4.2.5. Generalization test on InSAR data

InSAR is one important application of PU. However, public InSAR data is scarce. For generalization test, we generate InSAR data using the elevation maps collected from the Internet and form the wrapped data following Sec. 4.1. See Table 5 for the NRMSE results, where PUTFormer achieves the best results in 4/5 cases and produces better unwrapped images. The superior performance of PUTFormer is also reflected in the visual results provided in Fig. 3. All these comparisons have demonstrated the better generalization

Table 5: NRMSE results on InSAR data.

Method	60	20	10	5	0	Average
PURNN [23]	12.23	12.47	12.60	13.17	14.22	12.94
PhaseNet2.0 [29]	12.64	12.64	12.87	12.82	13.52	12.90
SQD-LSTM [24]	8.85	8.99	8.99	9.29	10.19	9.26
EESANet [32]	27.79	27.53	27.20	28.08	29.99	28.12
SwinIR [71]	17.44	17.35	17.3	17.51	17.86	17.49
Uformer [61]	6.35	6.36	6.38	6.42	7.06	6.51
Restormer [60]	<u>5.69</u>	5.82	<u>5.93</u>	<u>5.94</u>	**6.53**	<u>5.98</u>
PUTFormer	**5.51**	**5.54**	**5.53**	**5.87**	<u>6.97</u>	**5.88**

of PUTFormer to phase data from a different domain, compared to others.

4.2.6. Application on Fringe Projection Profilomeries

An important application of PU, Fringe Projection Profilometry (FPP), is included to further assess the performance of the proposed method in practice. In this experiment, we use the dataset proposed in [73] for this evaluation. The NRMSE results are presented in Table 6, which shows that the proposed method performs competitively in this application. Visual comparisons in Figure 4 demonstrate that the proposed method yields globally more accurate results than the other approaches.

It is interesting to see that for FPP data, classification-based methods (e.g., PhaseNet2.0) in general outperform most regression-based methods. One likely reason is that FPP data has high signal-to-noise ratio, in comparison to the data simulated in the other experiments. For the data with high signal-to-noise-ratio, predicting the wrap count will be very robust which

Table 6: NRMSE results on FPP data

Metric	PURNN [23]	PhaseNet2.0 [29]	SQD-LSTM [24]	EESANet [32]	SwinIR [71]	Uformer [61]	Restormer [60]	PUTFormer
NRMSE	5.31	<u>3.12</u>	3.91	4.36	5.47	3.76	3.48	**2.59**

lead to more accurate results that directly predicting unwrapped phase. Despite the advantage of classification-based methods on data with high signal-to-noise ratio, our method, a regression-based method, remains to be very compeititive against classification-based methods.

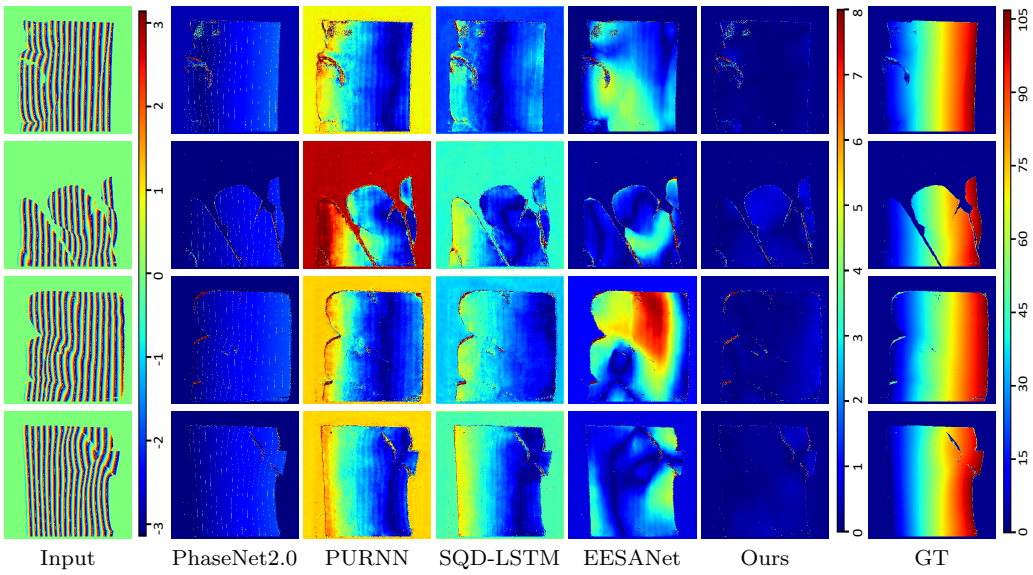

Figure 4: Visual inspection of residual images of Fringe Projection Profilomeries.

4.2.7. Comparison of computational cost

Table 7 compares different methods in terms of the number parameters and the test time on a 256×256 phase image. PUTFormer has both the second smallest model size and the shortest inference time. These advan-

Table 7: Comparison in computational complexity.

Metric	PURNN [23]	PhaseNet2.0 [29]	SQD-LSTM [24]	EESANet [32]	SwinIR [71]	Uformer [61]	Restormer [60]	PUTFormer
#Para (M)	1.07	1.15	**0.90**	23.88	0.90	20.60	11.72	<u>1.03</u>
Time (ms)	323.71	23.05	<u>15.62</u>	24.73	295.18	69.28	177.51	**10.04**

Table 8: NRMSE results in ablation studies.

Settings	w/o GTB	w/o LTB	w/o GTB & LTB	w/o PE	w/o CSM	Original
#1	5.23	1.75	12.47	2.28	1.60	1.19
#2	4.93	2.70	6.02	3.63	2.23	1.75

tages not only show its higher practical value, but also demonstrate that its superiority is from a better architecture design, not an increase of model complexity. Particularly, PUTFormer is much smaller and faster than Restomer while showing better performance in previous experiments. This has demonstrated the superiority of our DNN architecture design.

4.2.8. Visual comparison with other transformer models

To better demonstrate the superior performance of our PUTFormer over other transformer models, we show three examples in Fig. 5 to compare PUTFormer with SwinIR, Uformer, and Restormer. It can be seen that our PUTFormer shows advantages on handling phase images with dense phase changes, in comparison to other transformer models.

4.3. Ablation Studies

We construct and retrain several variants of PUTFormer for ablation studies. *(i)* w/o GTB: Discarding all GTBs. *(ii)* w/o LTB: Discarding all

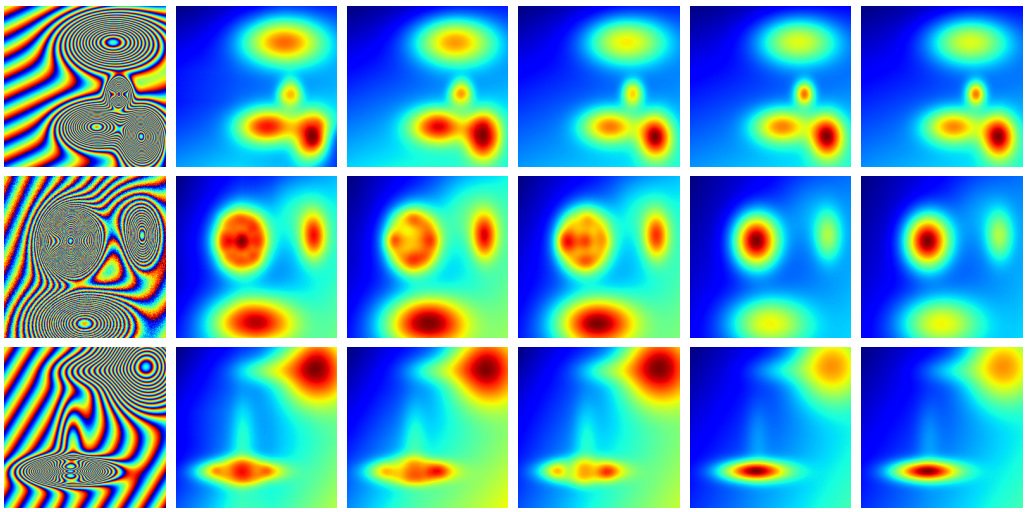

Figure 5: Visual comparison with transformer variants.

LTBs. *(iii)* w/o GTB & LTB: Replacing all GTBs and LTBs by standard 3×3 convolutional layers, resulting in a U-shaped CNN. *(iv)* w/o PE: Removing PE from all GTBs and LTBs. *(iv)* w/o CSM: Removing all CSMs. The feature channel numbers of these variants are adjusted to maintain the model size for fair comparison.

Two settings are tested: Setting #1 follows Section 4.2.1-1 with SNR=-2; and Setting #2 follows Section 4.2.4-4 with SNR=0. From the results listed in Table 8, we make the following remarks. *(i)* The GTBs capturing global spatial dependencies play a crucial role to the accuracy of PU. See also Fig. 6 for a visual example, where non-local distortion occurs when GTBs are removed. *(ii)* The LTBs also have noticeable contribution to the performance. As shown in the visual example of Fig. 6, LTBs bring local refinement during unwrapping. *(iii)* A pure CNN without GTBs and LTBs performs much

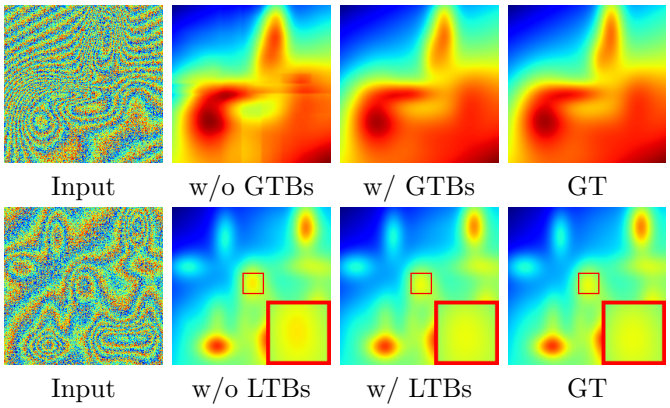

Figure 6: Results of PUTFormer w/ or w/o GTBs/LTBs.

worse. *(iv)* PE is quite useful, even more beneficial than LTBs. This is mainly due to that spatial order provides natural and informative constraints for PU. *(iv)* The CSMs are effective in fusing features of local and global semantics, leading to further performance gain.

We vary the model size of PUTFormer by changing the channel number in each module. See Table9 for the results of these variants on the two settings used above. We observe that even with a smaller model with a half size, our PUTFormer still achieves promising results and performs better than the most competitive PU-dedicated method, SQD-LSTM, as well as the most competitive transformer model, Restormer. Furthermore, increasing the model size leads to further gain in performance, yet not significant.

4.4. Visualization and Analysis on Attention

We use Fig. 7 to visualize the attention maps at different heads of multi-head attention in two GTBs of PUTFormer. As the number of self-attention maps is very large, we only show the ones regarding the center, *i.e.*, each

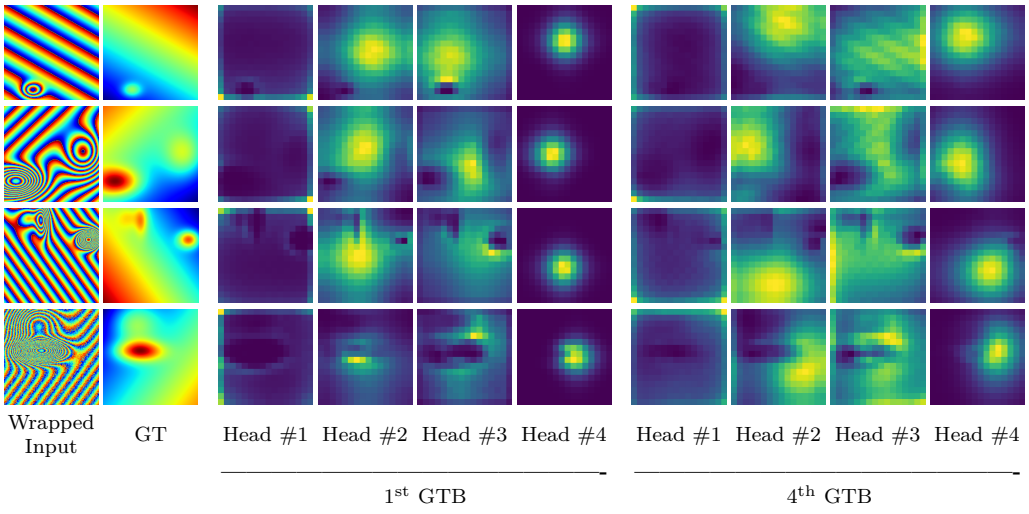

Figure 7: Visualization of self-attention maps (regarding the central point) generated by different attention heads of two GTBs.

attention map encodes the attention strength on each spatial location when unwrapping the phase at the central point. We have the following observations. *(i)* The heads within the same GTB capture different types of spatial dependencies. For instance, the attention maps produced by Head #1 of both GTBs focus on boundaries while the ones by Head #2 focus on inner regions. *(ii)* The attention performs selection on regions to unwrap. For instance, there are holes in the attention maps produced by Head #3. These holes correspond to the wrapped cluster regions with frequent wrap count changes, which are unrelated and even harmful to the unwrapping at the center point. Using the produced attention maps can bypass these regions. *(iii)* The attention maps tend to have more-global structures at the latter GTB.

Table 9: Performance of PUTFormer with varied model size.

Setting	0.58M	1.03M (Original)	1.60M	SQD-LSTM	Restormer
#1	1.97	1.19	1.04	3.27	1.56
#2	2.66	1.75	1.59	6.08	2.78

4.5. Limitation Analysis

The statistical characteristics of measurement noise of phase wrapping varies across different scenarios. For instance, in InSAR, the electronic noise in radar systems is typically close to Gaussian noise, whereas in optical interferometry, imperfections or dust on optical components can introduce spike noise. For supervised learning methods, optimal performance is achieved when the network is trained on data with noise characteristics similar to those of the testing data. A mismatch between the noise distributions in the training and testing data often leads to a noticeable decrease in generalization performance. While one could train the network on multiple datasets with varying noise distributions, its performance would significantly worse than the same network but trained specifically on data with a noise distribution that matches the testing data. Our approach also shares this limitation: to achieve the best performance, it requires prior knowledge of the noise distribution in the testing data to construct training samples with matching noise characteristics. This limitation is well known among supervised methods for image processing. In future work, we will explore how to efficiently adapt a model trained for one specific noise distribution to another.

5. Conclusion

In this paper, we proposed a transformer model tailored for PU, which is capable of capturing and exploiting rich global spatial dependencies within a phase image for unwrapping. Leveraging an efficient coarse-to-fine multi-resolution analysis architecture, our proposed model achieved noticeable performance gain over existing PU-dedicated DNNs and a popular transformer-based DNN, while using a lightweight model. Our future work will study further improvement on the generalization performance on unseen phase patterns.

References

[1] F. Yang, T.-A. Pham, N. Brandenberg, M. P. Lütolf, J. Ma, M. Unser, Robust phase unwrapping via deep image prior for quantitative phase imaging, IEEE Transactions on Image Processing 30 (2021) 7025–7037.

[2] M. Gontarz, V. Dutta, M. Kujawińska, W. Krauze, Phase unwrapping using deep learning in holographic tomography, Optics Express 31 (12) (2023) 18964–18992.

[3] H. Zhou, C. Cheng, H. Peng, D. Liang, X. Liu, H. Zheng, C. Zou, The phu-net: A robust phase unwrapping method for mri based on deep learning, Magnetic Resonance in Medicine 86 (6) (2021) 3321–3333.

[4] S. Zhang, Recent progresses on real-time 3d shape measurement using digital fringe projection techniques, Optics and lasers in engineering 48 (2) (2010) 149–158.

[5] J. Yu, F. Da, Absolute phase unwrapping for objects with large depth range, IEEE Transactions on Instrumentation and Measurement (2023).

[6] F. Peng, X. Zheng, Q. Miao, Large dynamic range and anti-fading phase-sensitive otdr using 2d phase unwrapping via neural network, IEEE Transactions on Instrumentation and Measurement (2023).

[7] H. An, Y. Cao, Y. Zhang, H. Li, Phase-shifting temporal phase unwrapping algorithm for high-speed fringe projection profilometry, IEEE Transactions on Instrumentation and Measurement 72 (2023) 1–9.

[8] Y. Wang, H. Xu, H. Zhu, X. Chen, Y. Wang, Pixel-wise phase unwrapping with adaptive reference phase estimation for 3-d shape measurement, IEEE Transactions on Instrumentation and Measurement 72 (2023) 1–9.

[9] J. Zeng, W. Ma, W. Jia, Y. Li, H. Li, X. Liu, M. Tan, Self-unwrapping phase-shifting for fast and accurate 3-d shape measurement, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–12.

[10] D. A. Ausherman, A. Kozma, J. L. Walker, H. M. Jones, E. C. Poggio, Developments in radar imaging, IEEE Transactions on Aerospace and Electronic Systems (4) (1984) 363–400.

[11] M. D. Pritt, Phase unwrapping by means of multigrid techniques for interferometric sar, IEEE Transactions on Geoscience and Remote Sensing 34 (3) (1996) 728–738.

[12] C. V. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, P. A. Thompson, Spotlight-mode synthetic aperture radar: a signal processing approach, Springer Science & Business Media, 2012.

[13] M. Takeda, K. Mutoh, Fourier transform profilometry for the automatic measurement of 3-d object shapes, Applied optics 22 (24) (1983) 3977–3982.

[14] X. Su, W. Chen, Fourier transform profilometry:: a review, Optics and lasers in Engineering 35 (5) (2001) 263–284.

[15] S. Zhang, S.-T. Yau, High-resolution, real-time 3d absolute coordinate measurement based on a phase-shifting method, Optics Express 14 (7) (2006) 2644–2649.

[16] W.-Y. Chang, F.-H. Hsu, K.-H. Chen, J.-H. Chen, K. Y. Hsu, Heterodyne moiré surface profilometry, Optics express 22 (3) (2014) 2845–2852.

[17] C. Li, Y. Cao, C. Chen, Y. Wan, G. Fu, Y. Wang, Computer-generated moiré profilometry, Optics Express 25 (22) (2017) 26815–26824.

[18] L. Wang, Y. Cao, C. Li, Y. Wan, H. Li, C. Xu, H. Zhang, Improved computer-generated moiré profilometry with flat image calibration, Applied Optics 60 (5) (2021) 1209–1216.

[19] M. Lu, X. Su, Y. Cao, Z. You, M. Zhong, Modulation measuring profilometry with cross grating projection and single shot for dynamic 3d shape measurement, Optics and Lasers in Engineering 87 (2016) 103–110.

[20] M. Zhong, X. Su, W. Chen, Z. You, M. Lu, H. Jing, Modulation measuring profilometry with auto-synchronous phase shifting and vertical scanning, Optics express 22 (26) (2014) 31620–31634.

[21] Z. Wei, Y. Cao, H. Wu, C. Xu, G. Ruan, F. Wu, C. Li, Dynamic phase-differencing profilometry with number-theoretical phase unwrapping and interleaved projection, Optics Express 32 (11) (2024) 19578–19593.

[22] J. Zhong, Y. Zhang, Absolute phase-measurement technique based on number theory in multifrequency grating projection profilometry, Applied optics 40 (4) (2001) 492–500.

[23] K. Ryu, S.-M. Gho, Y. Nam, K. Koch, D.-H. Kim, Development of a deep learning method for phase unwrapping mr images, in: Proceedings of the International Society for Magnetic Resonance in Medicine, Vol. 27, 2019, p. 4707.

[24] M. V. Perera, A. De Silva, A joint convolutional and spatial quad-directional lstm network for phase unwrapping, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2021, pp. 4055–4059.

[25] K. Wang, Q. Kemao, J. Di, J. Zhao, Deep learning spatial phase unwrapping: a comparative review, Advanced Photonics Nexus 1 (1) (2022) 014001.

[26] G. Dardikman, N. T. Shaked, Phase unwrapping using residual neural networks, in: Computational Optical Sensing and Imaging, Optica Publishing Group, 2018, pp. CW3B–5.

[27] Y. Qin, S. Wan, Y. Wan, J. Weng, W. Liu, Q. Gong, Direct and accurate phase unwrapping with deep neural network, Applied Optics 59 (24) (2020) 7258–7267.

[28] K. Wang, Y. Li, Q. Kemao, J. Di, J. Zhao, One-step robust deep learning phase unwrapping, Optics Express 27 (10) (2019) 15100–15115.

[29] G. Spoorthi, R. K. S. S. Gorthi, S. Gorthi, Phasenet 2.0: Phase unwrapping of noisy data based on deep learning approach, IEEE Transactions on Image Processing 29 (2020) 4862–4872.

[30] J. Zhang, X. Tian, J. Shao, H. Luo, R. Liang, Phase unwrapping in optical metrology via denoised and convolutional segmentation networks, Optics Express 27 (10) (2019) 14903–14912.

[31] T. Zhang, S. Jiang, Z. Zhao, K. Dixit, X. Zhou, J. Hou, Y. Zhang, C. Yan, Rapid and robust two-dimensional phase unwrapping via deep learning, Optics Express 27 (16) (2019) 23173–23185.

[32] J. Zhang, Q. Li, Eesanet: edge-enhanced self-attention network for two-dimensional phase unwrapping, Optics Express 30 (7) (2022) 10470–10490.

[33] G. Spoorthi, S. Gorthi, R. K. S. S. Gorthi, Phasenet: A deep convolutional neural network for two-dimensional phase unwrapping, IEEE Signal Processing Letters 26 (1) (2018) 54–58.

[34] J. Li, C. Li, Q. Zhang, B. Wu, T. Liu, X. Lu, J. Di, L. Zhong, Multi-wavelength network: Predicted-illumination for phase unwrapping in

quantitative phase imaging, Optics & Laser Technology 167 (2023) 109781.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (2017).

[36] R. M. Goldstein, H. A. Zebker, C. L. Werner, Satellite radar interferometry: Two-dimensional phase unwrapping, Radio science 23 (4) (1988) 713–720.

[37] D. J. Bone, Fourier fringe analysis: the two-dimensional phase unwrapping problem, Applied optics 30 (25) (1991) 3627–3632.

[38] M. A. Herráez, D. R. Burton, M. J. Lalor, M. A. Gdeisat, Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path, Applied optics 41 (35) (2002) 7437–7444.

[39] X. Su, W. Chen, Reliability-guided phase unwrapping algorithm: a review, Optics and Lasers in Engineering 42 (3) (2004) 245–261.

[40] H. Jiang, Y. Xu, C. Zhang, Z. Xu, J. Huang, H. Tan, J. Lu, An algorithm combining the branch-cut method and rhombus phase unwrapping algorithm, in: Journal of Physics: Conference Series, Vol. 1634, IOP Publishing, 2020, p. 012068.

[41] C. Xu, Y. Cao, H. Wu, H. Li, H. Zhang, H. An, Curtain-type phase unwrapping algorithm, Optical Engineering 61 (4) (2022) 044103–044103.

[42] D. C. Ghiglia, M. D. Pritt, Two-dimensional phase unwrapping: theory, algorithms, and software, Wiely-Interscience (1998).

[43] X. Xie, Iterated unscented kalman filter for phase unwrapping of interferometric fringes, Optics Express 24 (17) (2016) 18872–18897.

[44] D. Blinder, H. Ottevaere, A. Munteanu, P. Schelkens, Efficient multiscale phase unwrapping methodology with modulo wavelet transform, Optics Express 24 (20) (2016) 23094–23108.

[45] H. Y. Huang, L. Tian, Z. Zhang, Y. Liu, Z. Chen, G. Barbastathis, Path-independent phase unwrapping using phase gradient and total-variation (tv) denoising, Optics Express 20 (13) (2012) 14075–14089.

[46] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, D. Psaltis, M. Unser, Isotropic inverse-problem approach for two-dimensional phase unwrapping, JOURNAL OF THE OPTICAL SOCIETY OF AMERICA A 32 (6) (2015) 1092–1100.

[47] L. Bian, X. Wang, D. Li, Q. Ren, D. Zheng, Robust phase unwrapping via non-local regularization, Optics Letters 48 (6) (2023) 1399–1402.

[48] P. Wang, T. Peng, S. Zhang, F. Lu, Z. Zhong, J. Li, Y. Wang, J. Zhou, A phase unwrapping method with the sparse prior for diffraction phase microscopy, Optics & Laser Technology 170 (2024) 110268.

[49] X. Zhang, H. Wang, H. Peng, H. Du, Y. Jiao, S. Li, J. Zhang, Z. Pan, H. Huang, Y. Ju, A dsspi phase unwrapping method for improving the detection efficiency of cfrp-reinforced concrete defect, Optics & Laser Technology 168 (2024) 109862.

[50] L. Jiaying, X. Xianming, Central difference information filtering phase unwrapping algorithm based on deep learning, Optics and Lasers in Engineering 163 (2023) 107484.

[51] X. He, D. Zheng, Q. Kemao, G. Christopoulos, Quaternary gray-code phase unwrapping for binary fringe projection profilometry, Optics and lasers in engineering 121 (2019) 358–368.

[52] D. Zheng, Q. Kemao, F. Da, H. S. Seah, Ternary gray code-based phase unwrapping for 3d measurement using binary patterns with projector defocusing, Applied optics 56 (13) (2017) 3660–3665.

[53] Y. Wang, S. Zhang, Novel phase-coding method for absolute phase retrieval, Optics letters 37 (11) (2012) 2067–2069.

[54] Y. Xing, C. Quan, C. Tay, A modified phase-coding method for absolute phase retrieval, Optics and Lasers in Engineering 87 (2016) 97–102.

[55] H. An, Y. Cao, H. Li, H. Zhang, Temporal phase unwrapping based on unequal phase-shifting code, IEEE Transactions on Image Processing 32 (2023) 1432–1441.

[56] J. Wang, Y. Cao, H. Wu, Z. Wei, Absolute phase retrieval based on fringe amplitude encoding without any additional auxiliary pattern, Optics Express 31 (25) (2023) 41952–41966.

[57] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[58] X. Luo, W. Song, S. Bai, Y. Li, Z. Zhao, Deep learning-enabled invalid-point removal for spatial phase unwrapping of 3d measurement, Optics and Laser Technology 163 (2023) 109340.

[59] T. H. Kim, M. S. Sajjadi, M. Hirsch, B. Scholkopf, Spatio-temporal transformer network for video restoration, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 106–122.

[60] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, Restormer: Efficient transformer for high-resolution image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5728–5739.

[61] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, H. Li, Uformer: A general u-shaped transformer for image restoration, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 17683–17693.

[62] J. Xiao, X. Fu, A. Liu, F. Wu, Z.-J. Zha, Image de-raining transformer, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[63] F.-J. Tsai, Y.-T. Peng, Y.-Y. Lin, C.-C. Tsai, C.-W. Lin, Stripformer: Strip transformer for fast image deblurring, in: European Conference on Computer Vision, Springer, 2022, pp. 146–162.

[64] M. Shen, H. Gan, C. Ning, Y. Hua, T. Zhang, Transcs: a transformer-based hybrid architecture for image compressed sensing, IEEE Transactions on Image Processing 31 (2022) 6991–7005.

[65] M. Li, Y. Fu, Y. Zhang, Spatial-spectral transformer for hyperspectral image denoising, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 1368–1376.

[66] M. Li, J. Liu, Y. Fu, Y. Zhang, D. Dou, Spectral enhanced rectangle transformer for hyperspectral image denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5805–5814.

[67] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[68] N. Shazeer, Glu variants improve transformer, arXiv preprint arXiv:2002.05202 (2020).

[69] 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: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[70] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3202–3211.

[71] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, R. Timofte, Swinir: Image restoration using swin transformer, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 1833–1844.

[72] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[73] S. Bai, X. Luo, K. Xiao, C. Tan, W. Song, Deep absolute phase recovery from single-frequency phase map for handheld 3d measurement, Optics Communications 512 (2022) 128008.