SC-PNC: Semantic Communication-Empowered Physical-layer Network Coding

Haoyuan Pan, *Member, IEEE,* Shuai Yang, *Graduate Student Member, IEEE,* Tse-Tin Chan, *Member, IEEE,* Zhaorui Wang, Victor C. M. Leung, *Life Fellow, IEEE,* Jianqiang Li, *Member, IEEE*

***Abstract*—This paper puts forth the first framework for semantic communication (SC)-empowered physical-layer network coding (PNC), referred to as SC-PNC. Although conventional bit-oriented PNC can enhance the throughput of wireless relay networks by turning mutual wireless interference into useful network-coded information, it faces two primary problems that limit its application in practice. First, bit-oriented PNC decoding is susceptible to the relative phase offsets among signals received from different nodes; in particular, some "bad" relative phase offsets could lead to significant performance degradation. Second, the scheduling design of bit-oriented PNC transmissions is limited by the bitwise operation. To address these issues, this paper designs SC-PNC, which leverages semantic communication to bypass the need for bit-perfect message recovery at the destination. First, we employ a two-way relay network (TWRN) to demonstrate how SC-PNC effectively mitigates the detrimental effects of "bad" relative phase offsets. Then, we explore a triangular relay network (TriRN) to show how we can take advantage of semantic communication to redesign the scheduling of PNC transmissions. Specifically, an SC-PNC TriRN architecture is designed, wherein each node receives information from the other two nodes in only two time slots. Taking image delivery as an example, experimental results reveal that SC-PNC consistently achieves high and stable image reconstruction quality under different channel conditions and relative phase offsets, outperforming the conventional bit-oriented counterparts. Moreover, the new two-slot SC-PNC TriRN architecture is effective in extracting semantically accurate information from images, showcasing its potential as a low-latency solution for semantic information exchange.**

***Index Terms*—Semantic communication, physical-layer network coding (PNC), deep learning, PNC scheduling.**

This work was supported in part by the National Natural Science Foundation of China under Grants 62371302, 62325307, 62203134, and 62293482, in part by Shenzhen Science and Technology Program under Grants RCBS20210609103234060 and ZDSYS20220527171400002, in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603, in part by the Guangdong "Pearl River Talent Plan" under Grant 2019JC01X235, in part by Shenzhen Talents Special Project-Guangdong Provincial Innovation and Entrepreneurship Team Supporting Project under Grant 2021344612, in part by the Tencent "Rhinoceros Bird"-Scientific Research Foundation for Young Teachers of Shenzhen University, in part by the Basic Research Project under Grant HZQB-KCZYZ-2021067 of Hetao Shenzhen-Hong Kong Science and Technology Cooperation Zone, and in part by the Faculty Development Scheme under Grant UGC/FDS14/E02/21 and the Research Matching Grant Scheme from the Research Grants Council of Hong Kong. *(Corresponding author: Jianqiang Li.)*

This article was presented in part at the IEEE Wireless Communications and Networking Conference (IEEE WCNC), Mar. 2023.

H. Pan and S. Yang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China (e-mails: hypan@szu.edu.cn, 2110276206@email.szu.edu.cn).

T.-T. Chan is with the Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong SAR, China (e-mail: tsetinchan@eduhk.hk).

Z. Wang is with the Future Network of Intelligence Institute, and the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China (e-mail: wangzhaorui@cuhk.edu.cn).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada V6T 1Z4 (e-mail: vleung@ieee.org).

J. Li is with the National Engineering Laboratory for Big Data System Computing Technology, and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China (e-mail: lijq@szu.edu.cn).

I. Introduction

With the exponential growth in the number of intelligent devices, wireless communications between these smart devices have generated an unprecedented amount of data, making wireless spectrum resources a significant bottleneck in achieving efficient information exchange. As a key technique to effectively increase spectrum efficiency and enhance the throughput of wireless networks, physical-layer network coding (PNC) turns superimposed electromagnetic waves, i.e., wireless interference, into network-coded information [1], [2], which has garnered significant attention in various communication scenarios over the past decades [1].

A simple representation of PNC is seen in a two-way relay network (TWRN) [1], as shown in Fig. 1(a). In a TWRN, two end nodes, nodes A and B, want to send packets to each other over a wireless medium. Due to the long distance between nodes A and B or their low transmit power, there is no direct signal path between them. They need to communicate with the assistance of a relay node R. The advantage of PNC over traditional store-and-forward relaying is evident: it halves the time slots needed for packet exchange, i.e., from four to two time slots [2]. In the first time slot, the two nodes simultaneously send packets C^A and C^B to the relay in the same frequency band. In the second time slot, the relay performs PNC decoding on the superimposed received signals and broadcasts back a network-coded packet $C^A \oplus C^B$ to the nodes. We refer to the network-coded packet $C^A \oplus C^B$ as a PNC packet, an eXclusive-OR (XOR) of the two source packets from the end nodes. Upon receiving the PNC packet, the two nodes subtract their self-packet from the PNC packet to obtain the packet from another node. As a result, PNC halves the transmission time and doubles the throughput of a TWRN.

Conventional PNC studies were focused on the traditional bit-oriented communication paradigm, aiming at the rapid and reliable delivery of exact data bits. For example, each node in a TWRN aims to receive the exact bits from another node, and the relay tries to decode a bitwise XOR packet of the two

source packets. However, this paper argues that conventional bit-oriented PNC faces the following two primary problems that limit its use in practice.

- **Problem 1: Bit-oriented PNC decoding is affected by the relative phase offset among wireless signals of simultaneously transmitted packets.** Successful bitwise XOR decoding at the relay is critical to the system performance of all PNC-enabled networks. Taking a TWRN as an example, prior works revealed that the bit-oriented PNC decoding performance is adversely affected by the relative phase offset between the signals of simultaneously transmitted packets from the two nodes [2], [3]. Specifically, different relative phase offsets lead to different bit error rate (BER) performances in PNC decoding. As will be further elaborated in Section III-C, under "bad" relative phase offsets, when mapping the superimposed signals to network-coded symbols in PNC decoding, referred to as *PNC mapping*, some constellation points mapped to different network-coded symbols are inevitably overlapped with each other. This leads to symbol ambiguity in the symbol detection process and thus degrades the BER performance. While some approaches have been proposed to mitigate the relative phase offset issue, they are all bit-oriented solutions (i.e., minimizing BERs) that cannot solve the problem completely, even in the absence of noise (see Section II for the detailed articulation).
- **Problem 2: The scheduling design of PNC transmissions in bit-oriented relay networks is limited by the bitwise operation.** To see this, let us consider a PNC-enabled triangular relay network (TriRN), where three nodes A, B, and C want to send packets to other nodes via a relay R. That is, each node aims to receive information from the other two nodes. To achieve this, a total of four time slots are needed, instead of two time slots as a PNC-enabled TWRN does. If the scheduling of PNC transmissions in the TWRN were used in the TriRN, the first time slot would be an uplink slot where all the three nodes send simultaneously. Suppose the relay decodes a PNC packet $C^A \oplus C^B \oplus C^C$ and broadcasts it back to the users in the downlink slot. The PNC packet $C^A \oplus C^B \oplus C^C$ does not help to recover packets from other nodes. In general, due to the limitation of the bitwise operation, side information is required to utilize PNC packets to recover missing native packets. Side information could be the self-packet of a node, or packets obtained in different time slots. As a result, the conventional scheduling of PNC transmissions in a TriRN is shown in Fig. 1(b). In the first time slot, nodes A and B transmit packets C^A and C^B simultaneously. The relay receives and decodes a PNC packet $C^A \oplus C^B$. In the second time slot, nodes B and C transmit packets C^B and C^C simultaneously. The relay receives the second PNC packet $C^B \oplus C^C$. In the third and fourth time slots, the relay broadcasts $C^A \oplus C^B$ and $C^B \oplus C^C$, respectively. Now node A can extract C^B using its own packet C^A and $C^A \oplus C^B$. After recovering C^B, C^C can be obtained using $C^B \oplus (C^B \oplus C^C)$. Nodes B and C follow a similar process to extract two packets from the other two nodes using their own packets and the two broadcast PNC packets [4]. While the above scheduling design improves throughput over the store-and-forward relaying (in which six time slots are needed in total), the design of packet transmissions in bit-oriented PNC-enabled relay networks is bottlenecked by the bitwise operation.

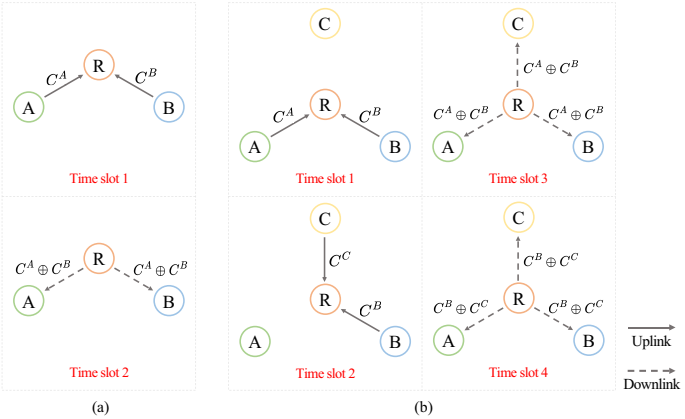


Fig. 1: (a) A bit-oriented PNC-enabled two-way relay network: two end nodes A and B exchange information via a relay node R; (b) A bit-oriented PNC-enabled triangular relay network: each node aims to receive information from the other two nodes via a relay node R.

Enabled by advanced deep learning technologies, semantic communication has recently attracted significant attention [5]. Various semantic communication systems based on deep neural networks (DNNs) were designed for the transmission of text [6], image [7], and speech signals [8]. In these works, the "semantics" of the source information is sent to the receiver, which relates to the content and meaning of the transmitted messages, instead of the exact bit stream transmission. Hence, traditional performance metrics such as BER are no longer applicable in semantic communication that evaluates system performance from a semantic level. Motivated by the recent advance of semantic communication, this paper puts forth the first framework for semantic communication (SC)-empowered PNC, referred to as SC-PNC, to tackle the limitations of conventional bit-oriented PNC schemes.

- To solve the relative phase offset problem (**Problem 1**), we use TWRN as an example to demonstrate that SC-PNC can effectively mitigate the detrimental effects of "bad" relative phase offsets. The key motivation is that a degraded BER performance does not mean no information is received by the receiver. This encourages us to extract the semantic meaning of transmitted messages directly instead of ensuring accurate bit stream transmission, thereby mitigating the impact of relative phase offsets on the PNC performance. As will be presented in Section IV, we design the first semantic communication-empowered PNC-enabled TWRN, referred to as SC-PNC TWRN. We jointly design DNN-based transceivers at the end nodes and propose a *semantic PNC decoder* at

the relay. With the help of DNNs, our designed SC-PNC TWRC realizes semantic PNC decoding at the relay and the direct extraction of semantic information at the end nodes. SC-PNC solves the performance degradation problem in the conventional bit-oriented communication design that only aims to deliver bit streams reliably.

- To overcome the bitwise limitation in the scheduling of bit-oriented PNC transmissions (**Problem 2**), we use TriRN as an example to show that with SC-PNC, we can design lower-latency semantic communication-empowered scheduling of PNC transmissions, hence significantly improving communication efficiency. The key motivation is that the scheduling design in SC-PNC does not need to follow the bitwise operation since semantic communication does not require destinations to recover every bit of the transmitted messages from sources. Moreover, since only semantic meaning is required, mutual wireless interference among nodes is tolerable to some extent. Therefore, new semantic communication-empowered PNC scheduling and transmission can be designed for a TriRN. As will be detailed in Section V, we design SC-PNC TriRN, a DNN-based PNC-enabled architecture for TriRN, where each node receives semantic information from the other two nodes using only *two* time slots, thus further reducing communication latency in semantic meaning exchange.

We employ image delivery as an application example to validate the effectiveness of SC-PNC. Through an extensive set of experiments, experimental results reveal that SC-PNC TWRN can achieve a high and stable peak signal-to-noise ratio (PSNR), a metric that evaluates the reconstruction quality of images [5], [9], despite under different signal-to-noise ratios (SNRs) and relative phase offsets. Furthermore, we show that the new two-slot SC-PNC TriRN architecture is effective in extracting semantically accurate information from images, even when impaired by wireless channels, highlighting its potential for low-latency semantic communication.

To conclude, we have the following major contributions:

(1) We put forth the first framework for semantic communication (SC)-empowered PNC, referred to as SC-PNC. SC-PNC TWRN is used as an example to demonstrate that SC-PNC can effectively mitigate the detrimental effects of bad relative phase offsets among signals received from different nodes.
(2) We put forth SC-PNC TriRN to show that with SC-PNC, low-latency semantic communication-empowered scheduling of PNC transmissions can be designed, thus overcoming the bitwise limitation in the scheduling of bit-oriented PNC transmissions.
(3) We validate the effectiveness of SC-PNC using image delivery as an example. SC-PNC is shown to achieve high and stable image reconstruction quality under different channel conditions and relative phase offsets, outperforming the conventional bit-oriented counterparts.

II. Related Work

A. Physical-layer Network Coding (PNC)

PNC has received significant attention in recent decades as it can turn mutual interference between wireless signals from simultaneous transmitters into useful network-coded information. PNC was first proposed to increase the throughput of a TWRN in [1]. Later, the idea of PNC was extended to various communication models, including both complex relay networks (e.g., [4]) and non-relay networks (e.g., wireless multiple access networks in [3], [10]). Under different network topologies, the scheduling of PNC transmissions was carefully designed, and PNC was shown to improve system performance under different communication metrics, such as network latency [4] and age of information (AoI) [11].

As described in Section I, the relative phase offset issue in PNC decoding affects the performance of the overall system in all PNC-enabled networks. Several approaches have been proposed to deal with the effect of "bad" relative phase offsets. For example, [12] designed special PNC mapping rules for the bad relative phase offsets such that the overlapping constellation points are mapped to the same network-coded symbol. However, such specially designed PNC mapping rules only apply to the non-channel-coded case and cannot be easily extended to the channel-coded case. With channel codes, certain advanced channel decoding methods were proposed in [2], [13], [14] to mitigate the effect of bad relative phase offsets. However, they are generally complex iterative decoding methods that induce significant decoding latency, which is detrimental in practical implementation. Non-iterative decoding schemes were investigated in [10], [15], [16], which however was limited to the binary phase shift keying (BPSK) modulation only. Moving beyond BPSK, [3] proposed a non-iterative multiple-antenna receiver for high-order modulations to increase the degree of freedom of the relative phase offset. The PNC decoding performance can be improved because it is not likely that both antennas experience a bad relative phase offset. Recently, deep learning (DL)-based approaches have been proposed to solve the network-coded symbol ambiguity problem [17], [18]. They trained an end-to-end TWRC using deep neural networks (DNNs), and a DNN-trained PNC mapping rule was shown to improve the BER performance. Nevertheless, the DNN-trained PNC mapping rule was learned from a dataset with only a 0° relative phase offset. As our experiments in Section VI point out, the system performance still degrades when the actual relative phase offset differs from 0°.

All the above prior works focused on the conventional bit-oriented communication paradigm to design and implement PNC-enabled networks. In contrast, this paper puts forth a semantic communication (SC)-empowered PNC framework, referred to as SC-PNC. We show that our SC-PNC can not only solve the relative phase offset problem, but also motivate new PNC scheduling methods, hence significantly improving communication efficiency.

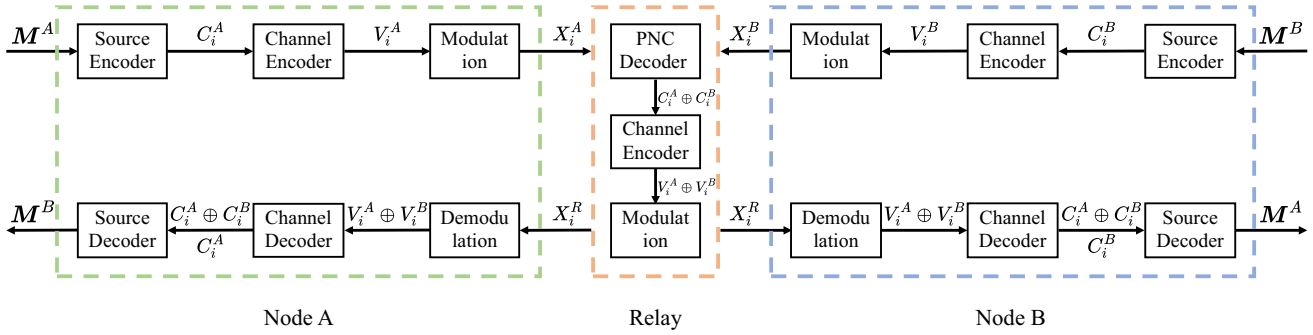


Fig. 2: The general architecture of a conventional bit-oriented PNC-enabled TWRN.

B. Semantic Communication

The essence of semantic communication is to deliver the "meanings" of information. Aided by advanced DL techniques, the semantics of messages are extracted and interpreted by well-trained DNNs at both the transmitter and the receiver [5]. Prior studies on semantic communication have been focused on point-to-point communication systems, aiming to deliver different types of messages, such as texts [6], [19], images [7], [20], speeches [8], [21], videos [22], and a combination of them, i.e., multi-modal data [23]. These works typically trained a joint source-channel coding (JSCC) by DNNs in an end-to-end manner. The DL-based JSCC methods were shown to significantly improve the adaptability of semantic communication to various channel conditions. In particular, it was shown that the semantic information can be "interpreted" successfully at the receiver even without receiving the complete and accurate bit information, especially when the SNR is low [24].

Besides point-to-point communication systems, semantic communication was also studied in various multiuser systems. For example, [25] proposed a one-to-many semantic communication system, which uses the semantic features of different users to build semantic recognizers based on pre-trained models to distinguish different users, and [26] proposed a task-oriented multiuser semantic communication framework to deliver both single-modal and multi-modal data. However, these works did not consider relay networks. Refs. [27], [28] studied simple one-way relay networks and compared the semantic information delivery performance under different relay forwarding methods. Our work also considers relay networks and focuses explicitly on PNC-enabled relay networks, i.e., PNC-enabled TWRNs and TriRNs.

In general, most works on semantic communication focused extensively on developing advanced DL models (i.e., to build a common semantic knowledge base at transceivers) to achieve semantic information delivery. These works do not take into account the advantage of semantic communication to redesign wireless communication networks. Since semantic information can be interpreted even without receiving the complete and accurate bit information, our work tries to take this advantage to design new communication network protocols. Specifically, we use a PNC-enabled TriRN as an example to show that we can actually design the scheduling of SC-PNC transmission so that communication efficiency can be further improved.

III. Preliminaries

A. Bit-Oriented PNC-Enabled Two-Way Relay Network (TWRN)

We first review the general information processing in a conventional bit-oriented PNC-enabled TWRN, as shown in Fig. 2. We consider nodes A and B want to exchange messages $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$, i.e., node A sends $\boldsymbol{M}^A$ to node B, and node B sends $\boldsymbol{M}^B$ to node A. As an example, messages $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$ are images in this paper. Each node tries to recover the image from the other node.

To begin with, large messages $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$ are source-encoded into multiple packets C_i^A and C_i^B, $i = 1, 2, \ldots$, respectively, i.e., the source encoding compresses the original message into binary bits, from which the original message can be recovered exactly. Let us consider a time-slotted system in which a packet C_i^A or C_i^B occupies one time slot. Next, channel coding adds redundant bits for detecting and correcting bit errors in wireless transmissions. Let $\Gamma(\cdot)$ denote the channel encoding operator, and we assume that linear channel codes are used in this paper. Then C_i^A and C_i^B are channel-encoded into $V_i^A = \Gamma(C_i^A)$ and $V_i^B = \Gamma(C_i^B)$, respectively. After channel coding, V_i^A and V_i^B are modulated into symbol sequences $X_i^A = (x_i^A[1], \ldots, x_i^A[k], \ldots)$ and $X_i^B = (x_i^B[1], \ldots, x_i^B[k], \ldots)$, respectively. $x_i^A[k]$ and $x_i^B[k]$ are the k-th modulated symbol of nodes A and B, respectively.

In an uplink slot of the TWRN, when nodes A and B transmit modulated symbols X_i^A and X_i^B to the relay simultaneously, the received superimposed signals $Y_i^R = (y_i^R[1], \ldots, y_i^R[k], \ldots)$ at the relay can be expressed as

$$Y_i^R = H_u^A \odot X_i^A + H_u^B \odot X_i^B + N^R, \tag{1}$$

where $H_u^A = (h_u^A[1], \ldots, h_u^A[k], \ldots)$ and $H_u^B = (h_u^B[1], \ldots, h_u^B[k], \ldots)$ are the uplink channel coefficients of the two nodes with respect to the relay, $\odot$ represents the Hadamard product, $N^R \sim \mathcal{CN}\left(0, \sigma^2 \mathbf{I}\right)$ indicates the independent and identically distributed (i.i.d.) complex Gaussian noise vector with variance σ^2 at the relay, and $\mathbf{I}$ denotes a vector with the same length as X_i, where each element is 1.[1] The PNC decoder at the relay attempts to decode the linear

[1]Since conventional PNC-enabled TWRNs are usually implemented via orthogonal frequency-division multiplexing (OFDM) [2], [16], here we employ an OFDM system where multipath fading can be dealt with by the cyclic prefix (CP) of OFDM symbols.

combination of C_i^A and C_i^B, denoted by $C_i^A \oplus C_i^B$, from the superimposed signal Y_i^R. In other words, $C_i^A \oplus C_i^B$ is the eXclusive-OR (XOR) of C_i^A and C_i^B, referred to as a PNC packet.

In the subsequent downlink slot, the relay channel-encodes $C_i^A \oplus C_i^B$ into $\Gamma(C_i^A \oplus C_i^B) = \Gamma(C_i^A) \oplus \Gamma(C_i^B) = V_i^A \oplus V_i^B$ (note: $\Gamma(\cdot)$ is linear). Then $V_i^A \oplus V_i^B$ is modulated into X_i^R, which is broadcast to nodes A and B. The received signals at node A or B is

$$Y_i^K = H_d^K \odot X_i^R + N^K,\ K \in \{A, B\}, \tag{2}$$

where H_d^K and N^K represent the downlink channel coefficient and noise term of the wireless link from relay R to node $K \in \{A, B\}$, respectively. The signals received at nodes A and B are demodulated and channel-decoded into $C_i^A \oplus C_i^B$. Using $C_i^A \oplus C_i^B$, node A obtains C_i^B by $C_i^B = C_i^A \oplus \left(C_i^A \oplus C_i^B\right)$. Node B follows the same manner to obtain $C_i^A = C_i^B \oplus \left(C_i^A \oplus C_i^B\right)$. Finally, after receiving all C_i^A and C_i^B, $i = 1, 2, \ldots$, original messages $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$ are recovered by source decoding.

B. Bit-Oriented PNC-Enabled Triangular Relay Network (TriRN)

Besides TWRNs, PNC can be applied to other relay networks. For example, [4] identifies nine fundamental building blocks of complex PNC networks, including the triangular relay network (TriRN). In a TriRN, each node aims to receive information from the other two nodes with the help of a relay. Taking node A as an example, it wants to receive messages $\boldsymbol{M}^B$ and $\boldsymbol{M}^C$ from nodes B and C, respectively. With the bit-oriented communication, as in a TWRN, a large message $\boldsymbol{M}^K,\ K \in \{A, B, C\}$, is first source-encoded into multiple packets C_i^K, $i = 1, 2, \ldots$. After channel coding and modulation, modulated symbols X_i^K are sent from node K.

In a bit-oriented PNC-enabled TriRN, each node receives two packets, one from each of the other two nodes, in a total of four time slots. More specifically, in the first time slot, nodes A and B transmit packets C_i^A and C_i^B simultaneously. The relay receives and decodes a PNC packet $C_i^A \oplus C_i^B$. In the second time slot, nodes B and C transmit packets C_i^B and C_i^C simultaneously. The relay receives the second PNC packet $C_i^B \oplus C_i^C$. In the third and fourth time slots, the relay broadcasts $C_i^A \oplus C_i^B$ and $C_i^B \oplus C_i^C$, respectively. To recover other users' packets, taking node A as an example, it extracts C_i^B using its own packet C_i^A and $C_i^A \oplus C_i^B$. After recovering C_i^B, C_i^C can be obtained using $C_i^B \oplus (C_i^B \oplus C_i^C)$. Nodes B and C follow a similar process to extract two packets from the other two nodes using their own packets and the two broadcast PNC packets. In contrast, without PNC, it is easy to figure out that a total of six time slots are needed: three for sending packets from the sources to the relay and three for forwarding packets from the relay to the destinations [4].

C. The Relative Phase Offset Issue in Bit-Oriented PNC-Enabled Communication

In the above examples of PNC-enabled TWRN and TriRN, we have not considered the possibility of PNC decoding failure at the relay. In practice, successful PNC decoding at the relay is critical to the overall system performance. Prior works have shown that conventional PNC decoding at the relay in the uplink phase is affected by the relative phase offset between the signals of the two simultaneously transmitted packets, especially when high-order modulations are used [3]. To see this, let us consider a TWRN as an example and assume the use of QPSK modulation. We focus on the k-th received superimposed symbol $y_i^R[k]$

$$y_i^R[k] = h_u^A[k]x_i^A[k] + h_u^B[k]x_i^B[k] + n[k], \tag{3}$$

where $x_i^A[k]$ and $x_i^B[k]$ are the k-th modulated QPSK symbols of nodes A and B, and $x_i^A[k],\ x_i^B[k] \in \{1+j, 1-j, -1+j, -1-j\}$. An important issue in PNC decoding is how to calculate $x_i^A[k] \oplus x_i^B[k]$ (abbreviated as $x_i^{A\oplus B}[k]$) using the received sample $y_i^R[k]$ in (3). This process is referred to as *PNC mapping*. According to [3], to maintain the linearity of linear channel codes $\Gamma(\cdot)$, the QPSK PNC mapping rule should be defined as

$$\begin{aligned} x_i^{A\oplus B}[k] &= x_i^{A,I}[k] \oplus x_i^{B,I}[k] + j \cdot (x_i^{A,Q}[k] \oplus x_i^{B,Q}[k]), \\ &= x_i^{A,I}[k]x_i^{B,I}[k] + j \cdot (x_i^{A,Q}[k]x_i^{B,Q}[k]), \end{aligned} \tag{4}$$

where $x_i^{A,I}[k],\ x_i^{A,Q}[k] \in \{-1, 1\}$ ($x_i^{B,I}[k]$, $x_i^{B,Q}[k]$) represents the real and imaginary part of $x_i^A[k]$ ($x_i^B[k]$), respectively.

To illustrate the effect of relative phase offset on PNC decoding conveniently, let us consider a noise-free case where nodes A and B have the same received power at the relay with uplink channel gains $h_u^A[k] = 1$ and $h_u^B[k] = e^{j\Delta\phi}$, and $\Delta\phi$ is the relative phase offset between the signals of the two nodes. Then the k-th superimposed symbol received at the relay is $y_i^R[k] = x_i^A[k] + x_i^B[k]e^{j\Delta\phi}$. Under some "bad" relative phase offsets, if the PNC mapping rule (4) is used, several constellation points mapped to different network-coded symbols inevitably overlap. For example, as pointed out in [3], when $\Delta\phi = 90°$, the constellation points of symbol pairs $(1-j, 1-j)$ and $(1+j, -1-j)$ overlap at 2, i.e., $(1-j) + (1-j)e^{j90°} = (1+j) + (-1-j)e^{j90°} = 2$, but they are mapped to $1+j$ and $-1-j$, respectively. This means that when $\Delta\phi = 90°$, the QPSK mapping rule in (4) can lead to symbol ambiguity, resulting in a degraded PNC BER performance, even in the absence of noise. A detailed BER performance comparison under different $\Delta\phi$ can be founded in [3].

D. Towards Semantic PNC-enabled Networks

In practice, the relative phase offset is random due to uncoordinated and distributed transmitters. The overall BER performance of the PNC decoder is limited by the worst-case relative phase offset. To deal with this issue, prior solutions focused on the conventional bit-oriented communication paradigm that could not completely eliminate the effect of random relative phase offsets. For example, although sophisticated iterative PNC decoding schemes [2], [13], [14] are possible to improve the BER performance under the "bad" relative phase offsets, such iterative schemes lead to high complexity and large latency, and therefore are not amenable

to practical implementation. Packet retransmission is another solution as it introduces diversity and it is only a small chance that all transmissions experience "bad" relative phase offsets in a real wireless environment. When PNC decoding fails at the relay, Automatic Repeat reQuest (ARQ) is often used so that the sources retransmit the previous packets [11]. Retransmission generally requires more time resources and is not applicable to low-latency applications as well. However, if ARQ is not used, maintaining packets in the presence of bit errors affects the quality of image recovery.

Motivated by the new semantic communication paradigm, Section IV designs the first semantic communication-empowered PNC-enabled TWRN, referred to as SC-PNC TWRN, to address the relative phase offset problem. Thanks to the jointly designed DNN-based transceivers at the end nodes and the semantic PNC decoder at the relay, we find that even with a bad relative phase offset (say, $90°$), the two nodes can exchange accurate meaning on a semantic level.

Following SC-PNC TWRN, Section V further puts forth a semantic communication-empowered PNC-enabled TriRN, referred to as SC-PNC TriRN. Since semantic communication does not require destinations to recover every bit of the transmitted messages from sources, new PNC scheduling and transmission can be designed to further reduce latency. We present a DNN-based PNC-enabled architecture for TriRN, where each node receives information from the other two nodes using only two time slots. Recall that even without packet loss, the bit-oriented PNC-enabled TWRN requires four time slots. Furthermore, by conveying semantics instead of all bits of a message, each node transmits less information in semantic communication than in conventional bit-oriented communication, which is favorable for low-latency communication.

IV. Semantic Communication-Empowered PNC – Part I: TWRN

In this section, we first introduce the overall framework of the proposed SC-PNC TWRN. After that, the DNN implementation details of SC-PNC TWRN are presented.

A. Overall Framework of SC-PNC TWRN

As in the conventional PNC-enabled TWRN, we consider that nodes A and B exchange messages $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$ (e.g., images in this paper) via relay R in SC-PNC TWRN. The system architecture, shown in Fig. 3, consists of three main components: (i) an encoder at each node, including a semantic encoder and a channel encoder, (ii) a semantic PNC decoder at relay R, and (iii) a decoder at each node, including a channel decoder and a semantic decoder. We detail these three components as follows.

1) The Encoder at the End Nodes: In an SC-PNC TWRN, $\boldsymbol{M}^A$ and $\boldsymbol{M}^B$ are encoded into symbol sequences X_{sc}^A and X_{sc}^B using DNNs, which are then sent by the nodes over the wireless links simultaneously. During the encoding procedure, two encoders are used at node A or B: the semantic encoder and the channel encoder. The semantic encoder extracts the main features (i.e., semantic information) of the message,

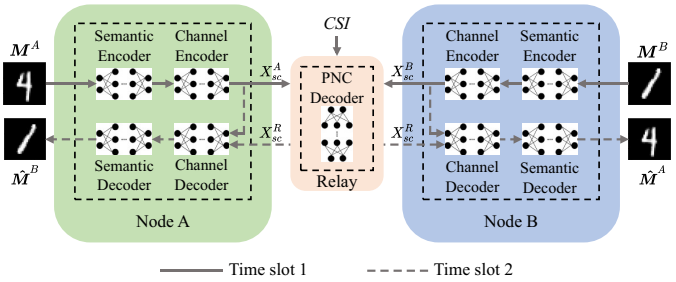


Fig. 3: The overall framework of SC-PNC TWRN.

while the channel encoder encodes the semantic information to mitigate wireless channel impairments and further compresses it into transmitted symbols. The detailed DNN implementation of the semantic encoder and the channel encoder will be explained in the next subsection.

Let us denote the training parameters of the DNNs for the semantic encoder and the channel encoder as α and β, respectively. Then the encoded symbol sequences X_{sc}^K, $K \in \{A, B\}$, is

$$X_{sc}^K = \mathbf{T}_\beta^c \left(\mathbf{T}_\alpha^s \left(\boldsymbol{M}^K \right) \right), \tag{5}$$

where $\mathbf{T}_\alpha^s(\cdot)$ denotes the semantic encoder with parameters α, and $\mathbf{T}_\beta^c(\cdot)$ denotes the channel encoder with parameters β.[2] As in the traditional PNC-enabled TWRN, nodes A and B transmit symbols X_{sc}^A and X_{sc}^B to the relay simultaneously in an uplink slot (see Fig. 3). We use Y_{sc}^R to denote the superimposed signals received by the relay in SC-PNC TWRN. At the relay, the semantic PNC decoding of the superimposed signals Y_{sc}^R is performed.

2) The Semantic PNC Decoder: Unlike the conventional PNC decoder discussed in Section III, a key difference of the semantic PNC decoder is that the relay does not perform a PNC mapping on the received superimposed signals according to specific PNC mapping rules (such as (4)). Instead, as shown in Fig. 3, we adopt a DNN as the PNC decoder for mapping the superimposed signals to the network-coded symbols broadcast in the downlink. In other words, the network-coded symbols broadcast back to the end nodes are directly obtained from the output of a well-trained DNN.

Prior work [29] pointed out that when training a DNN-based decoder (in fact, joint training of both encoder and decoder), the stochastic nature of the channel coefficient H in fading channels leads to the failure of the DNN model parameters to converge to the global optimum. As a result, the channel decoder cannot accurately recover the semantic information based on a local optimum. Hence, equalization should be performed to eliminate the effect of channel coefficients. For example, the channel coefficients can be estimated via a conventional least-squared estimator [30]. However, for PNC uplink, nodes A and B generally have different channel

[2]Note that since the whole TWRN (including the semantic and channel encoders in each node, the semantic and channel decoders in each node, and the PNC decoder at the relay) is trained in an end-to-end manner, the DNN-based semantic encoder and decoder of different users have different weights in their DNN models.

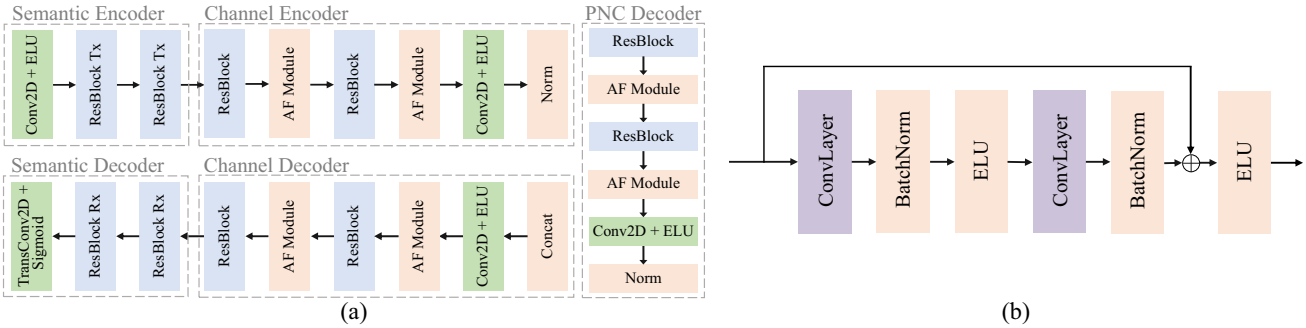


Fig. 4: (a) DNN implementation details in SC-PNC TWRN: for ease of illustration, here we only show the information processing flow from one node to another. The two-way information processing flow can be generalized accordingly. (b) The general architecture of a residual block.

coefficients, making it impossible to perform equalization for both nodes simultaneously. To tackle this issue, we input the channel coefficients of both nodes into the semantic PNC decoder for jointly learning how to eliminate the effects of the fading channels. Denote the DNN parameters of the semantic PNC decoder as δ. The symbols broadcast by the relay, X_{sc}^R, can be expressed as

$$X_{sc}^R = \mathbf{H}_\delta \left(Y_{sc}^R, H_u^A, H_u^B \right), \tag{6}$$

where $\mathbf{H}_\delta(\cdot)$ denotes the semantic PNC decoder with parameters δ, H_u^A and H_u^B are the uplink channel coefficients of the two nodes with respect to the relay. Nodes A and B receive signals Y_{sc}^A and Y_{sc}^B in the downlink, respectively.

3) The Decoder at the End Nodes: Similar to the encoder, the decoder also consists of two parts: the channel decoder and the semantic decoder. As in the traditional bit-oriented PNC-enabled TWRN, a node combines the network-coded symbols received from the relay with the symbols it sent to recover the symbols from the other node. In SC-PNC, the channel decoder performs the same procedure and obtains the semantic information of the message from the other node. The semantic decoder then reconstructs the message based on this semantic information. Both the channel decoder and semantic decoder are implemented using DNNs. Let η and φ denote their DNN parameters, respectively. The reconstructed message at node A (i.e., the message sent by node B) is expressed as

$$\hat{\boldsymbol{M}}^B = \mathbf{R}_\varphi^s \left(\mathbf{R}_\eta^c \left(X_{sc}^A, \tilde{Y}_{sc}^A \right) \right), \tag{7}$$

where $\mathbf{R}_\eta^c(\cdot)$ denotes the channel decoder with parameters η and $\mathbf{R}_\varphi^s(\cdot)$ denotes the semantic decoder with parameters φ. $\tilde{Y}_{sc}^A = Y_{sc}^A / H_d^A$ denotes the received symbols after equalization using the downlink channel coefficients H_d^A from the relay to node A. The reconstructed message at node B can be obtained in the same manner.

B. DNN Implementation Details in SC-PNC TWRN

We now describe the DNN implementation of each component in SC-PNC TWRN, as shown in Fig. 4. For ease of illustration, Fig. 4(a) only shows the information processing flow from one node to another. The two-way information processing flow can be generalized accordingly. First, a batch of b images, $\boldsymbol{M}^K \in \mathbb{R}^{h \times w \times c}, K \in \{A, B\}$, are input to the encoder for training the DNNs, where $\mathbb{R}$ denotes the set of real numbers. Here b is the batch size, h is the height of each image, w is the width of each image, and c corresponds to the number of channels of the image (i.e., $c = 1$ for grayscale images, and $c = 3$ for color images in RGB format). Each pixel value of the images is normalized to a range of $[0, 1]$.

TABLE I: Parameter Settings for the Semantic Enc/Decoder.

	ConvLayer	Parameter	Settings
Semantic Encoder*	Conv2D	filters	128,128; 256,256
		kernel size	3,3; 3,3
		stride	2,1; 2,1
Semantic Decoder	TransConv2D	filters	64,64; 128,128
		kernel size	3,3; 3,3
		stride	1,2; 1,2

* For example, since the semantic encoder has two ResBlocks Tx, each consisting of two Conv2D, the filter parameters corresponding to the four Conv2D are 128, 128, 256, and 256, respectively.

As shown in Fig. 4(a), the implementation of the semantic encoder starts with a two-dimensional convolutional (Conv2D) layer with an exponential linear unit (ELU) activation function, followed by two residual blocks. Fig. 4(b) shows the general architecture of a residual block. The ConvLayers in ResBlock Tx are Conv2D, and the detailed parameters of Conv2D in the two ResBlock Tx of the semantic encoder are summarized in Table I. The ResBlock Tx downsamples the source image and extracts the higher-level image features. After the semantic encoder, we can obtain a batch of b semantic features $f \in \mathbb{R}^{\frac{h}{4} \times \frac{w}{4} \times 256}$ of the images in our implementation.

To map the semantic feature f of the images to transmitted symbols, the channel encoder consists of several ResBlock and attention feature (AF) modules, where each ConvLayer in ResBlock has 256 filters with kernels of size 3×3 and a stride size of 1. The AF Modules are introduced in the channel encoder to improve robustness in different SNR regimes [31], which takes the SNR and semantic features as the input to produce channel-aware features. After passing through the Conv2D with an ELU activation function, we obtained a batch of b transmitted symbols $X_{sc}^K \in \mathbb{C}^{m \times 1}, K \in \{A, B\}$, by reshaping and mapping the output of the Conv2D to the real and imaginary parts of the transmitted symbols. Here $\mathbb{C}$ denotes

the set of complex numbers, and m is the total number of transmitted symbols. Following [32], we refer to the dimension of the source signal $h \times w \times c$ as the source bandwidth and the dimension of output symbols m as the channel bandwidth. The *channel bandwidth ratio* of SC-PNC systems is then defined as $\rho_{sc} \triangleq m/(h \times w \times c)$, i.e., a larger channel bandwidth ratio indicates that each pixel in the image occupies more channel resources. Finally, a normalization layer is required on transmitted symbols $X_{sc}^A(X_{sc}^B)$ to ensure a unit transmit power constraint, i.e., $\mathbb{E}\left\|X_{sc}^K\right\|^2 = 1,\ K \in \{A, B\}$.

When the relay receives a batch of b superimposed signals $Y_{sc}^R \in \mathbb{C}^{m\times 1}$, the real and imaginary parts of the received superimposed symbols are concatenated with the channel coefficients H^A and H^B and passed together to the semantic PNC decoder. With respect to Fig. 4(a), the semantic PNC decoder consists of several ResBlock and AF Modules. Each ConvLayer in the two ResBlock of the semantic PNC encoder has 256 filters with kernels of size 3×3 and a stride size of 1. After passing through a Conv2D layer with an ELU activation function and normalization layers, the relay obtains a batch of b network-coded symbols, $X_{sc}^R \in \mathbb{C}^{m\times 1}$.

At the end nodes, the channel decoder first concatenates symbols X_{sc}^K transmitted by node K and the received network-coded symbols Y_{sc}^K sent by the relay through a concatenate layer, which is then reshaped and passed through the Conv2D with an ELU activation function layer. Later, several ResBlock and AF Module are used, where each ConvLayer in a ResBlock also has 256 filters with a kernel size of 3×3 and a stride size of 1. After the channel decoder, we obtain a batch of b estimated semantic features $\hat{f} \in \mathbb{R}^{\frac{h}{4}\times\frac{w}{4}\times 256}$ of the images.

Images are recovered by passing $\hat{f}$ to the semantic decoder. Each ConvLayer in a ResBlock Rx is now a TransConv2D layer for upsampling the image features and recovering the size of the image. The parameters of TransConv2D in the two ResBlock Rx of the semantic decoder are summarized in Table I. Note that the activation function in the last TransConv2D of the semantic decoder is sigmoid instead of ELU in our implementation. Finally, the de-normalization layer rescales each value of the image pixels back to the range of $[0, 255]$. A batch of b images are reconstructed as $\hat{\boldsymbol{M}}^B(\hat{\boldsymbol{M}}^A) \in \mathbb{R}^{h\times w\times c}$.

C. Loss Function in DNN Training

In image delivery applications, the objective of SC-PNC-TWRN is to reconstruct images as accurately as possible, i.e., to minimize the average distortion between a transmitted image and its reconstructed one. Hence, we adopt the mean squared error (MSE) as the loss function in the DNN training. As an example, suppose that the original image and the reconstructed image are grayscale images with size $h \times w$, denoted by $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$ respectively. The MSE between $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$ is

$$MSE(\boldsymbol{M}, \hat{\boldsymbol{M}}) = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w}\left(\boldsymbol{M}_{i,j} - \hat{\boldsymbol{M}}_{i,j}\right)^2, \quad (8)$$

where $\boldsymbol{M}_{i,j}(\hat{\boldsymbol{M}}_{i,j})$ represents the pixel value at the position (i, j) of the image. The loss function in training SC-PNC TWRN is defined as

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left[MSE\left(\boldsymbol{M}_n^A, \hat{\boldsymbol{M}}_n^A\right) + MSE\left(\boldsymbol{M}_n^B, \hat{\boldsymbol{M}}_n^B\right)\right], \quad (9)$$

where $MSE(\boldsymbol{M}_n^A, \hat{\boldsymbol{M}}_n^A)$ $(MSE(\boldsymbol{M}_n^B, \hat{\boldsymbol{M}}_n^B))$ is the MSE between the n-th transmitted image $\boldsymbol{M}_n^A(\boldsymbol{M}_n^B)$ and the reconstructed image $\hat{\boldsymbol{M}}_n^A(\hat{\boldsymbol{M}}_n^B)$, and N is the number of images.

V. Semantic Communication-Empowered PNC – Part II: TriRN

As presented in Section III-B, a bit-oriented PNC-enabled TriRN requires four time slots in total for each node to receive two packets from the other two nodes. With the help of semantic communication, this section presents a new DNN-based TriRN architecture that requires only two time slots for each node to receive semantic information from the other two nodes. We refer to this new TriRN architecture as SC-PNC TriRN. After that, we present how to extend the DNN implementation from TWRN to TriRN.

A. DNN-based PNC-Enabled TriRN Architecture: SC-PNC TriRN

Following the idea of SC-PNC TWRN, we put forth SC-PNC TriRN where each node wants to receive semantic information from the other two nodes, e.g., node A wants to recover messages $\boldsymbol{M}^B$ and $\boldsymbol{M}^C$ from nodes B and C, respectively. SC-PNC TriRN requires only two time slots to complete such information exchanges among the three nodes, whose system architecture is shown in Fig. 5. As in SC-PNC TWRN, the first time slot is an uplink slot, in which the three nodes send messages to the relay simultaneously, and the relay maps the superimposed signals into network-coded information. In the second time slot, the relay broadcasts network-coded information to the three nodes, and each node tries to extract semantic information from the other two users via the broadcast network-coded information and its self-information.

The whole SC-PNC TriRN system also consists of three main components: the encoder at each node, the semantic PNC decoder at relay R, and the decoder at each node, as detailed in the following. In an uplink slot, messages $\boldsymbol{M}^A$, $\boldsymbol{M}^B$, and $\boldsymbol{M}^C$ are encoded into symbol sequences X_{sc}^A, X_{sc}^B, and X_{sc}^C through a semantic encoder and a channel encoder. We remark that in an uplink slot, the bit-oriented PNC-enabled TriRN allows only two out of the three nodes to transmit signals over the wireless channel simultaneously. However, SC-PNC TriRN allows all the three nodes to transmit signals simultaneously, as shown in Fig. 5. Hence, the received superimposed signals Y_{sc}^R at the relay in an SC-PNC TriRN uplink is

$$Y_{sc}^R = H_u^A \odot X_{sc}^A + H_u^B \odot X_{sc}^B + H_u^C \odot X_{sc}^C + N^R, \quad (10)$$

where H_u^A, H_u^B, and H_u^C are the uplink channel coefficients of the nodes with respect to the relay and N^R indicates the i.i.d. complex Gaussian noise vector.

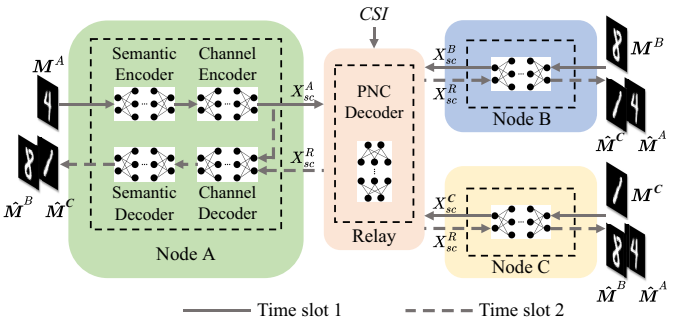


Fig. 5: The overall framework of SC-PNC TriRN. All nodes A, B, and C have the same encoder and decoder architecture, so only the detailed DNN architecture of node A is shown in the figure.

At the relay, as in SC-PNC TWRN, the network-coded symbols broadcast back to the end nodes are obtained from the output of the semantic PNC decoder, i.e., a well-trained DNN-based PNC mapping from the superimposed signals to the network-coded symbols. Note that since the superimposed signals Y_{sc}^R received by the relay R now contain information from the three nodes, the channel coefficients of all three nodes need to be fed into the semantic PNC decoder to jointly learn and obtain the network-coded symbols. Thus, the network-coded symbols broadcast by the relay, X_{sc}^R, in an SC-PNC TriRN can be reformulated as

$$X_{sc}^R = \mathbf{H}_\delta \left(Y_{sc}^R, H_u^A, H_u^B, H_u^C \right), \tag{11}$$

where $\mathbf{H}_\delta(\cdot)$ denotes the semantic PNC decoder in SC-PNC TriRN with parameters δ. Notice that for consistency, we use the same notation δ as in SC-PNC TWRN to denote the DNN parameters of the semantic PNC decoder in SC-PNC TriRN; in practice, the trained DNN parameters in the two architectures are different.

In the downlink slot, X_{sc}^R is broadcast to nodes A, B, and C, and they receive signals Y_{sc}^A, Y_{sc}^B, and Y_{sc}^C, respectively. Instead of recovering only one message from the opposite node as in SC-PNC TWRN, the decoder of each node in SC-PNC TriRN tries to recover two messages from the other two nodes (see Fig. 5). Specifically, let η and φ denote the DNN parameters of the channel decoder and semantic decoder, respectively. By combining the received network-coded information and its self-information, the two reconstructed messages sent by nodes B and C at node A can be expressed as

$$\left[\hat{M}^B, \hat{M}^C \right] = \mathbf{R}_\varphi^s \left(\mathbf{R}_\eta^c \left(X_{sc}^A, \tilde{Y}_{sc}^A \right) \right), \tag{12}$$

where $\tilde{Y}_{sc}^A$ denotes the received symbols after equalization, $\mathbf{R}_\eta^c(\cdot)$ denotes the channel decoder with parameters η and $\mathbf{R}_\varphi^s(\cdot)$ denotes the semantic decoder with parameters φ. Notice again that we use the same DNN parameter notations as in SC-PNC TWRN for consistency.

B. DNN Implementation From SC-PNC TWRN to SC-PNC TriRN

The DNN implementation of each component in SC-PNC TriRN is the same as that of SC-PNC TWRN, except for the setting of the last TransConv2D layer in the semantic decoder, as compared in Fig. 6(a) and Fig. 6(b). In SC-PNC TriRN, after passing through the channel decoder, we obtain a batch of b estimated mixture semantic features $\hat{f} \in \mathbb{R}^{\frac{h}{4} \times \frac{w}{4} \times 256}$ for the images from the other two nodes. The semantic decoder then reconstructs both messages from the other two nodes based on this "mixed" semantic information. As shown in Fig. 6(b), the semantic decoder in SC-PNC TriRN also contains two ResBlock Rx and one TransConv2D (the parameters of the two ResBlock Rx are summarized in Table I). Since the semantic decoder of each node now needs to recover two images simultaneously, as shown in Fig. 6(b), the last TransConv2D of the semantic decoder is set to $2c$ convolutional filters, comparing to c in Fig. 6(a) of SC-PNC TWRN, where c corresponds to the number of channels of the image. Finally, the semantic decoder in SC-PNC TriRN outputs a $h \times w \times 2c$ dimensional tensor that is reshaped into two separate images with a size of $h \times w \times c$. The de-normalization layer rescales each value of the image pixels back to the range of $[0, 255]$.

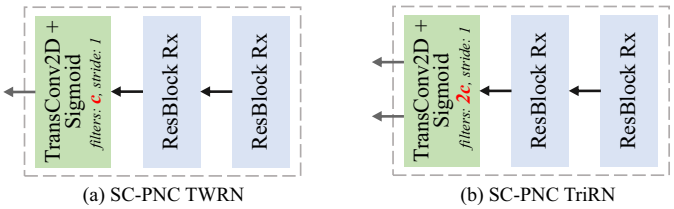


Fig. 6: The semantic decoder implementation details in (a) SC-PNC TWRN, and (b) SC-PNC TriRN: the differences lie in the parameter setting of the filters in the last TransConv2D layer.

VI. Experimental Evaluation

This section evaluates the performances of our proposed SC-PNC schemes. Specifically, the image recovery performance is evaluated by PSNR, a metric that measures the reconstruction quality of images. Section VI-A first presents the setup for our experiments, including the training details and benchmarking schemes. Section VI-B examines how SC-PNC solves the relative phase offset problem compared to the conventional bit-oriented schemes, using SC-PNC TWRN as an example. Section VI-C demonstrates that SC-PNC TriRN can achieve low-latency communication with a high PSNR in both additive white Gaussian noise (AWGN) and block Rayleigh fading channels. In addition to the PSNR evaluation, Section VI-D further considers object classification tasks on images to better evaluate the semantic information recovery performance.

A. Experimental Setup

To evaluate the performance of SC-PNC TWRN and SC-PNC TriRN, we consider using the MNIST handwritten digit dataset [33] in our experiments. The MNIST dataset consists of $70,000$ images ($60,000$ for training and $10,000$ for testing) of handwritten digits (zero to nine). The MNIST digits are grayscale images, each with 28×28 pixels, and each pixel is represented by a single intensity value in the range 0 (black) to 1 (white).

We first explain the implementation of our proposed SC-PNC TWRN. We implement SC-PNC TWRN using TensorFlow 2.8 [34]. The Adam optimizer [35] is adopted to train the DNNs. We set the training batch size to $b = 128$ and fix the learning rate to 0.001. For each training batch, each node sends b MNIST training images to the other node. Both AWGN and block Rayleigh fading channels are considered in the experiments. The uplink and downlink transmissions between a node and the relay are assumed to have an equal SNR. In addition, nodes A and B have the same SNR with respect to the relay, which is uniformly generated between 0dB and 16dB. Hence, the overall training dataset for SC-PNC TWRN can be represented by $\{(\boldsymbol{M}_i^A, \boldsymbol{M}_i^B, H_{u,i}^A, H_{u,i}^B, SNR_i)\}_{i=1}^N$, where N is the total number of training image pairs. Notably, for AWGN channels, since $|H_{u,i}^A| = |H_{u,i}^B| = 1$, it is equivalent to having a training data set $\{(\boldsymbol{M}_i^A, \boldsymbol{M}_i^B, \Delta\phi_i^{AB}, SNR_i)\}_{i=1}^N$, i.e., the relative phase offset $\Delta\phi^{AB}$ between nodes A and B in a TWRN is uniformly generated between 0° and 360° for each pair of transmitted images $\boldsymbol{M}_i^A$ and $\boldsymbol{M}_i^B$ during the training process.

For SC-PNC TriRN, we use the same DNN hyperparameters as in SC-PNC TWRN. To construct the training dataset, the SNR is uniformly generated between 0dB and 16dB in both uplink and downlink transmissions. Therefore, the overall training dataset for SC-PNC TriRN can be represented by $\{(\boldsymbol{M}_i^A, \boldsymbol{M}_i^B, \boldsymbol{M}_i^C, H_{u,i}^A, H_{u,i}^B, H_{u,i}^C, SNR_i)\}_{i=1}^N$, where N is the total number of training image triples. For AWGN channels, the training dataset is reduced to $\{(\boldsymbol{M}_i^A, \boldsymbol{M}_i^B, \boldsymbol{M}_i^C, \Delta\phi_i^{AB}, \Delta\phi_i^{BC}, SNR_i)\}_{i=1}^N$, where $\Delta\phi^{AB}$ ($\Delta\phi^{BC}$) denote the relative phase offset between nodes A and B (nodes B and C). (Note: the relative phase offset between nodes A and C, $\Delta\phi^{AC}$, can be derived from $\Delta\phi^{AB}$ and $\Delta\phi^{BC}$ by $\Delta\phi^{AC} = (\Delta\phi^{AB} + \Delta\phi^{BC})\ mod\ 360$, where mod denotes the modulo operator.) In other words, $\Delta\phi^{AB}$ or $\Delta\phi^{BC}$ is uniformly generated between 0° and 360° for each triple of transmitted images $\boldsymbol{M}_i^A$, $\boldsymbol{M}_i^B$, and $\boldsymbol{M}_i^C$.

For benchmarking purposes, we consider the following two schemes:

(1) **Conv-PNC**: This is the conventional PNC-enabled TWRN or TriRN described in Section III.[3] The rate-$1/2$ $[133, 171]_8$ convolutional codes defined in the IEEE 802.11 standards [16] are used for channel coding. QPSK is used (i.e., two source bits per symbol). At the relay, the so-called XOR channel decoding (XOR-CD) [3] is used to decode the PNC packet. The XOR-CD decoder maintains the linearity of convolutional codes when performing the QPSK PNC mapping (4). To demonstrate the end-to-end performance of image recovery, when the relay fails to decode a PNC packet (e.g., the decoded packet does not pass the cyclic redundancy check), the relay will still forward the decoded packet in the presence of bit errors in the downlink. As such, the numbers of time slots for conventional PNC-enabled TWRN and TriRN are two and four, respectively, as described in Section III.

(2) **D-PNC**: In [17], a DNN-based PNC TWRN scheme (D-PNC) was proposed. Still, the D-PNC scheme is bit-oriented, aiming to minimize the BER of decoded PNC packets at the relay. We reproduce a D-PNC TWRN using the same hyperparameter settings described in [17]. A joint channel coding and modulation scheme is obtained by training the DNNs. Specifically, each symbol sent by the end nodes is a DNN-trained QPSK modulated symbol (instead of a regular QPSK modulated symbol in Conv-PNC), containing two source bits. Moreover, we follow [17] to set the relative phase offset $\Delta\phi$ to 0° in the training process. As will be discussed in Section VI-B, due to the bit-oriented nature of the D-PNC TWRN, the image recovery performance of the D-PNC TWRN is significantly degraded when the relative phase offset used in training differs from the one used in testing. Therefore, we do not extend the D-PNC TWRN in [17] to a D-PNC TriRN.

[3]For simplicity and comparison with D-PNC and SC-PNC, end nodes send images directly without source coding. An image is simply divided into several packets, followed by channel coding and modulation.

Since image delivery applications are usually concerned with how similar the reconstructed image is at the receiver compared with the transmitted image, we measure the performances of different systems in terms of PSNR, which measures the squared intensity difference between the reconstructed and original image pixels [9]. Specifically, for a grayscale image of size $h \times w$, the PSNR between the transmitted image $\boldsymbol{M}$ and the reconstructed image $\hat{\boldsymbol{M}}$ is defined as

$$PSNR = 10\log_{10}\left(\frac{(MAX_{\boldsymbol{M}})^2}{MSE(\boldsymbol{M}, \hat{\boldsymbol{M}})}\right)\ dB, \qquad (13)$$

where $MAX_{\boldsymbol{M}}$ is the maximum possible pixel value of the transmitted image, and $MSE(\boldsymbol{M}, \hat{\boldsymbol{M}})$ is the MSE between the transmitted image $\boldsymbol{M}$ and the reconstructed image $\hat{\boldsymbol{M}}$. Note that Conv-PNC and D-PNC directly use the received image, possibly with incorrect pixel values, as $\hat{\boldsymbol{M}}$ to calculate PSNR. If the MSE between the transmitted image and the reconstructed image is smaller, the PSNR is larger and the reconstruction quality of the image is better.

B. Performance Evaluation of SC-PNC TWRN

We first evaluate the performance of TWRNs under different schemes. Fig. 7 plots the PSNR of Conv-PNC, D-PNC, and SC-PNC TWRN under different SNRs, when the relative phase offset $\Delta\phi^{AB}$ is (a) 0°, (b) 45°, and (c) 90° in AWGN channels. The channel bandwidth ratio of SC-PNC TWRN is set to $\rho_{sc} = 1/3$. When testing the PSNR performances, we consider an SNR-balanced scenario in which nodes A and B have the same SNR in both uplink and downlink transmissions, varying from -3dB to 6dB. In addition, the PSNRs of the two end nodes are evaluated separately. For the legends in Fig. 7, as an example, SC-PNC$_{\mathrm{AB}}$(TWRN) represents SC-PNC TWRN when node A sends to node B, i.e., the PSNR is measured at node B.

Fig. 7 shows that the PSNR increases with SNR for all three systems. In particular, when the relative phase offset $\Delta\phi^{AB}$ is

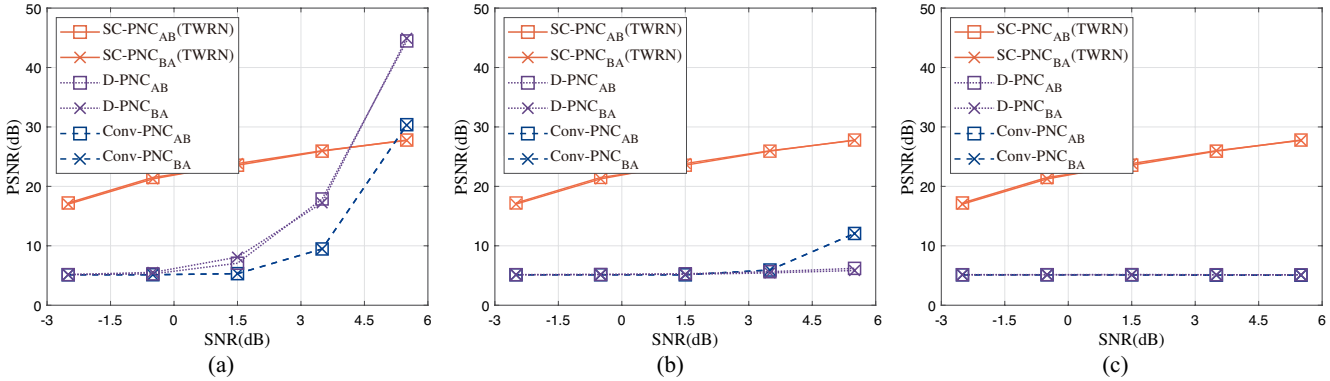


Fig. 7: PSNR performance of Conv-PNC, D-PNC, and our proposed SC-PNC TWRN under different SNRs when the relative phase offsets are (a) $0°$, (b) $45°$, and (c) $90°$ in AWGN channels, where SC-PNC TWRN transmits with a bandwidth ratio of $\rho_{sc} = 1/3$.

$0°$ (see Fig. 7(a)), D-PNC outperforms Conv-PNC because D-PNC jointly trains the channel coding and modulation scheme by deep learning and achieves lower BERs than the separately designed schemes in Conv-PNC. Furthermore, SC-PNC TWRN outperforms D-PNC when the SNR is low. At low SNRs, D-PNC suffers from high BERs due to poor channel conditions, resulting in low PSNRs. However, SC-PNC TWRN can still recover useful semantic information at low SNRs. In addition, D-PNC performs better than SC-PNC TWRN when the SNR is high, because D-PNC can reconstruct the transmitted image with accurate pixel values (i.e., without bit errors). By contrast, SC-PNC TWRN recovers images at the semantic level and does not obtain exact pixel values of the original image as D-PNC does. This phenomenon is consistent with previous works on semantic communication [7].

Fig. 7(b) and (c) show the PSNR performance against SNR when the relative phase offsets $\Delta\phi^{AB}$ are $45°$ and $90°$. As shown in both figures, Conv-PNC and D-PNC suffer from performance degradation compared with Fig. 7(a) when $\Delta\phi$ is $0°$. For Conv-PNC, the increase in $\Delta\phi^{AB}$ reduces the Euclidean distance between constellation points mapped to different XOR symbols, thus degrading the BER performance [3] (lowering the PSNR as well). For example, when $\Delta\phi^{AB}$ is $90°$, there are different XOR symbols overlapping each other with a Euclidean distance of zero. This leads to XOR symbol ambiguity when the QPSK PNC mapping rule (4) is used. For D-PNC, it only considers $\Delta\phi^{AB} = 0°$ during the training process. When the actual $\Delta\phi^{AB}$ is different from the one in training, the joint channel coding and modulation scheme learned from the training data does not work well. In other words, to achieve a good PSNR performance, the DNNs of D-PNC should be re-trained to facilitate different relative phase offsets.[4]

Fig. 7(b) and (c) show that SC-PNC TWRN outperforms both Conv-PNC and D-PNC under all SNR values. Compared with Conv-PNC (with the QPSK PNC mapping (4)), which leads to XOR symbol ambiguity when $\Delta\phi^{AB}$ is $90°$, SC-PNC TWRN does not rely on a specific PNC mapping rule and learns an appropriate rule via DNNs, thus avoiding the symbol ambiguity problem and improving the PSNR performance. While D-PNC also learns the PNC mapping via DNNs, it is designed based on minimizing BERs. When the trained $\Delta\phi^{AB}$ is different from the tested $\Delta\phi^{AB}$, an improper PNC mapping learned from DNNs leads to a degraded BER performance. With the new semantic communication paradigm, SC-PNC TWRN introduces the semantic encoder/decoder jointly trained with the channel encoder/decoder in the transceiver design. By doing so, messages are directly recovered at the semantic level instead of the bit level. At the semantic level, experimental results show that there is no need to worry about the disparity between the trained and the tested $\Delta\phi^{AB}$. For example, when we generate $\Delta\phi^{AB}$ randomly between $0°$ and $360°$ in the training process, Fig. 7 indicates that SC-PNC TWRN can achieve a high and stable PSNR under different SNRs and relative phase offsets.

C. Performance Evaluation of SC-PNC TriRN

We now evaluate the PSNR performance of SC-PNC TriRN. Let us first consider AWGN channels as shown in Fig. 8. In this experiment, the channel bandwidth ratio is set to $1/3$ for all SC-PNC systems. For the legends in Fig. 8, as those in Fig. 7, SC-PNC$_{\mathrm{BA}}$(TriRN) represents the PSNR performance when node A recovers the image from node B in SC-PNC TriRN.

The previous subsection has demonstrated that SC-PNC TWRN can achieve a stable PSNR performance under different relative phase offsets. Similarly, as indicated in Fig. 8, SC-PNC TriRN also has a stable PSNR under different relative phase offset pairs ($\Delta\phi^{AB}$, $\Delta\phi^{BC}$). Specifically, Fig. 8 plots the PSNRs of SC-PNC TriRN in two scenarios when (1) $\Delta\phi^{AB} = \Delta\phi^{BC} = 90°$ and (2) $\Delta\phi^{AB} = 0°$, $\Delta\phi^{BC} = 90°$. We observe that the PSNRs in the two scenarios are almost identical when node A recovers the images sent by nodes B and C. Thus, we omit the results of other relative phase offset

[4]To be fair, we also train D-PNC by generating $\Delta\phi^{AB}$ uniformly between $0°$ and $360°$ for each pair of $\boldsymbol{M}_i^A$ and $\boldsymbol{M}_i^B$ as SC-PNC does. However, this approach leads to poor BER/PSNR performances under all $\Delta\phi^{AB}$. This indicates that a unified joint channel coding and modulation scheme aimed at low BERs cannot be learned by DNNs when the relative phase offset varies. In contrast, SC-PNC performs well as it recovers the semantic meaning of the images instead of the exact bit information.

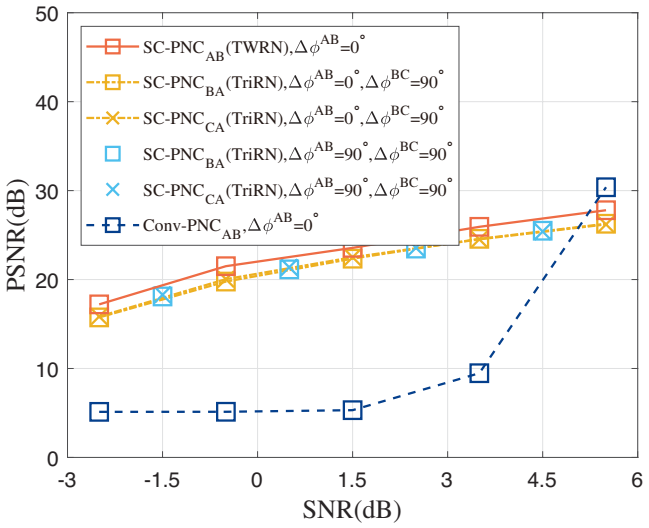


Fig. 8: PSNR performances of SC-PNC TriRN, SC-PNC TWRN and Conv-PNC in AWGN channels, where SC-PNC schemes transmit with a bandwidth ratio of $\rho_{sc} = 1/3$.

pairs in Fig. 8 (as well as in subsequent AWGN results). In addition, since each node in SC-PNC TriRN has the same encoder/decoder structure and all nodes have the same SNR, we only consider the image recovery performance at node A for clear presentation.

Fig. 8 plots the PSNRs of Conv-PNC schemes as benchmarks. In our experimental results, the Conv-PNC TriRN and TWRN have almost the same PSNR performance. In fact, the PSNR performance of the Conv-PNC TriRN is no better than its TWRN counterpart. Recall the four-slot scheduling of a Conv-PNC TriRN described in Section III-B. Considering node A, if the relay fails to decode $C_i^A \oplus C_i^B$ but still forwards the PNC packet in the presence of bit errors in the downlink, the recovered C_i^B will also contain bit errors (i.e., node A tries to extract C_i^B by using its own packet C_i^A and the forwarded $C_i^A \oplus C_i^B$). This further affects the recovery of C_i^C when using the error-prone C_i^B and $C_i^B \oplus C_i^C$ (even if $C_i^B \oplus C_i^C$ is error-free). Thus, the PSNR performance of node A in a Conv-PNC TriRN is worse than that in a Conv-PNC TWRN. Now considering node B, when it tries to recover C_i^A and C_i^C, whether the recovery of C_i^A (using its own packet C_i^B and the forwarded $C_i^A \oplus C_i^B$) is error-free does not affect the recovery of C_i^C (using its own packet C_i^B and the forwarded $C_i^B \oplus C_i^C$). Hence, the PSNR performance of node B in Conv-PNC TriRN is the same as that in Conv-PNC TWRN. Overall, the PSNR performance of the Conv-PNC TriRN is upper-bounded by the PSNR performance of its corresponding TWRN. Therefore, in Fig. 8, we use Conv-PNC TWRN to represent the PSNR performance of both schemes.

Furthermore, it is possible to use the SC-PNC TWRN to implement an SC-PNC TriRN following the conventional four-time-slot scheduling, which we call the four-slot SC-PNC TriRN. Following the same discussion about the Conv-PNC TriRN and TWRN above, the PSNR performance of the four-slot SC-PNC TriRN is no better than the PSNR performance of SC-PNC TWRN. Thus, we use SC-PNC TWRN to represent the PSNR performance of both schemes as benchmarks in Fig. 8.

Comparing the achieved PSNR of SC-PNC TriRN with its benchmarks, Fig. 8 shows that SC-PNC TriRN generally achieves a better PSNR performance than Conv-PNC, which is consistent with the TWRN results in the previous subsection. Moreover, SC-PNC TriRN can achieve a similar PSNR performance to SC-PNC TWRN. As demonstrated in Fig. 8, the PSNR achieved by SC-PNC TriRN is slightly lower than SC-PNC TWRN. This is because each node in SC-PNC TriRN now needs to recover two images of the other two nodes simultaneously, where the signals of the other two nodes interfere with each other. This indicates that mutual interference among nodes is tolerable to some extent because only semantic meaning is required, which can be extracted by DNNs. Moreover, as presented in Section V-B, the DNN implementation of SC-PNC TriRN is the same as that of SC-PNC TWRN, except for the setting of the last TransConv2D layer in the semantic decoder. The similar PSNR performance eases practical system designs where different network topologies are possible.

In Fig. 8, each node in SC-PNC TriRN transmits an image with a bandwidth ratio of $\rho = 1/3$, i.e., an MNIST image is sent via $\lfloor 28 \times 28 \times 1/3 \rfloor = 261$ symbols. In Conv-PNC, one pixel is represented by an eight-bit integer. After channel coding and modulation, each image is encoded using 6272 symbols. This shows that SC-PNC TriRN consumes fewer channel resources and less airtime to send an image than Conv-PNC, but achieves a significantly better PSNR. More importantly, compared with the four-slot SC-PNC TriRN, a two-slot SC-PNC TriRN where all the three nodes send simultaneously in the first time slot further reduces the time required to exchange images for each node while obtaining similar PSNR performances. Therefore, our new two-slot SC-PNC TriRN architecture is a viable solution to low-latency semantic communications.

We now present the performances of different schemes under Raleigh fading channels, where the PSNRs of SC-PNC TriRN, SC-PNC TriRN, and Conv-PNC versus SNR are plotted in Fig. 9(a). We observe that SC-PNC TriRN can still achieve similar PSNR performance as SC-PNC TWRN over the entire SNR range. For Conv-PNC, the random phase offsets in the channel gains under Rayleigh fading channels exacerbate the relative phase offset problem. SC-PNC TriRN and SC-PNC TWRN significantly outperform Conv-PNC in PSNR, indicating that our SC-PNC schemes can also extract useful semantic meanings under Raleigh fading channels. Furthermore, Fig. 9(b) compares the original and reconstructed images under different transmission schemes when the SNR is 9dB. We see in Fig. 9(b) that it is difficult to identify the digits on the reconstructed images delivered by Conv-PNC. By contrast, we can easily recognize the digits on the reconstructed images delivered by SC-PNC, although the exact pixel values in the reconstructed images may not be the same as the original ones. Such visualization results are consistent with the PSNR performances plotted in Fig 9(a).

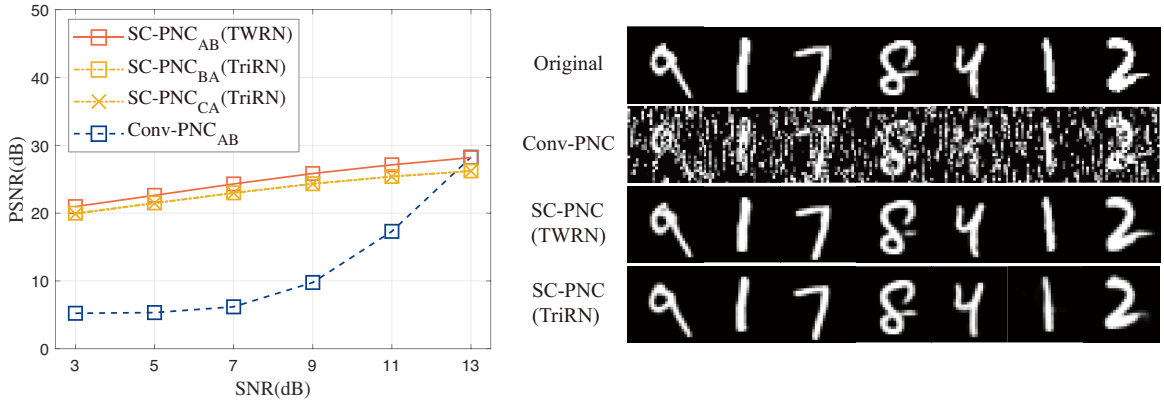


Fig. 9: (a) PSNR performances of SC-PNC TWRN, SC-PNC TriRN, and Conv-PNC under Rayleigh fading channels, where SC-PNC schemes have a channel bandwidth ratio of $\rho_{sc} = 1/3$. (b) Visualization of MNIST image samples for Conv-PNC, SC-PNC TWRN, and SC-PNC TriRN delivered over a Rayleigh fading channel of 9dB SNR.

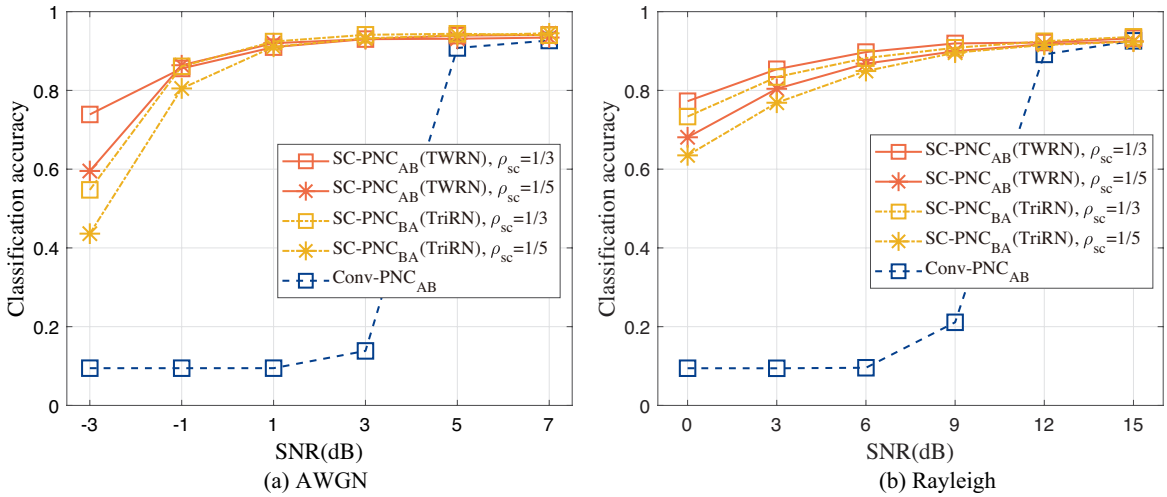


Fig. 10: Classification accuracy performance of different schemes in (a) AWGN channels (zero relative phase offsets) and (b) Rayleigh fading channels with different channel bandwidth ratios.

D. Performance of Image Classification

Previous subsections consider PSNR as the performance metric to compare the image reconstruction performance of different schemes. However, PSNR measures the MSE between the transmitted image and the reconstructed image only, i.e., a smaller MSE indicates a larger PSNR, so the reconstruction quality of the image is better. Hence, PSNR is unrelated to the true semantic information shown in images. In this subsection, we consider image classification and adopt classification accuracy as a performance metric for an image semantic communication system using the MNIST dataset. Specifically, we use a pre-trained vision transformer (ViT) [36] to recognize the digits on the reconstructed images, where the ViT classifies the digits by understanding image features so that the true semantic information of images can be extracted. We collect the classification accuracy based on the output of the ViT using the reconstructed images delivered by SC-PNC TriRN, SC-PNC TWRN, and Conv-PNC, respectively. A higher classification accuracy indicates better system performance.

Fig. 10(a) and (b) plot the classification accuracy of different schemes under AWGN channels and Rayleigh fading channels, respectively. In general, the classification accuracy is consistent with the PSNR performances presented in previous subsections. For example, we observe in Fig. 10 that the classification accuracy using the images delivered by Conv-PNC is low, especially when the SNR is small, which further confirms that Conv-PNC suffers from the relative phase offset problem and can hardly recover the semantic meaning of images. In contrast, the classification accuracy of SC-PNC TWRN and SC-PNC TriRN is much higher than that of Conv-PNC, even when a smaller channel bandwidth ratio $\rho_{sc} = 1/5$ is used. When the SNR is high, all the three schemes have almost the same classification accuracy. However, as explained earlier, the SC-PNC schemes consume fewer channel resources and less airtime than Conv-PNC to deliver the semantic information of an image. Overall, both the PSNR and classification accuracy metrics indicate that our proposed SC-PNC schemes

can extract accurate semantic information with low latency from images impaired by wireless channels.

VII. Conclusion

We have presented the first semantic communication-empowered PNC framework, referred to as SC-PNC, to tackle the limitations of conventional bit-oriented PNC schemes. Specifically, SC-PNC solves the relative phase offset problem in bit-oriented PNC decoding and overcomes the bitwise limitation in the scheduling of bit-oriented PNC transmissions.

Conventional bit-oriented PNC decoding is affected by the relative phase offsets among wireless signals of simultaneously transmitted packets. We use SC-PNC TWRN as an example to demonstrate the feasibility of using the new semantic communication paradigm to tackle the bad relative phase offset in PNC decoding. With the help of DNNs, our designed SC-PNC TWRN realizes semantic PNC decoding at the relay and direct extraction of semantic information at the end nodes. This solves the performance degradation problem in the conventional bit-oriented communication design that only aims to deliver bit streams reliably. Experiments on image delivery show that SC-PNC TWRN outperforms its bit-oriented benchmarks by achieving high and stable PSNR performance at different relative phase offsets.

Furthermore, while the scheduling design of conventional bit-oriented PNC transmissions is limited by the bitwise operation, we put forth SC-PNC TriRN to show that the scheduling of SC-PNC does not need to follow the bitwise operation. Since mutual wireless interference among nodes is tolerable to some extent when only semantic meaning is required, we design SC-PNC TriRN to allow each node to receive information from the other two nodes using only two time slots. Experimental results on PSNR and classification accuracy indicate that the new two-slot SC-PNC TriRN architecture extracts semantic information accurately with low latency from images impaired by wireless channels. Although this paper focuses on SC-PNC TriRN only, the insight of taking advantage of semantic communication to redesign wireless communication network protocols generally applies to other network typologies, which is a promising direction to further improve the performances of communication networks.

References

[1] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. ACM MobiCom*, Sep. 2006, pp. 358–365.

[2] S. C. Liew, L. Lu, and S. Zhang, "A primer on physical-layer network coding," *Synthesis Lectures on Communication Networks*, vol. 8, no. 1, pp. 1–218, Jun. 2015.

[3] H. Pan, L. Lu, and S. C. Liew, "Network-coded multiple access with high-order modulations," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9776–9792, Jun. 2017.

[4] J. He and S. C. Liew, "Building blocks of physical-layer network coding," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2711–2728, May 2015.

[5] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," Jun. 2022. [Online]. Available: https://arxiv.org/abs/2201.01389

[6] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep source-channel coding for sentence semantic transmission with HARQ," *IEEE Trans. Commun.*, vol. 70, no. 8, pp. 5225–5240, Jun. 2022.

[7] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, May 2019.

[8] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Jun. 2021.

[9] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[10] L. Lu, L. You, and S. C. Liew, "Network-coded multiple access," *IEEE Trans. Mob. Comput.*, vol. 13, no. 12, pp. 2853–2869, Apr. 2014.

[11] H. Pan, T.-T. Chan, V. C. M. Leung, and J. Li, "Age of information in physical-layer network coding enabled two-way relay networks," *IEEE Trans. Mob. Comput.*, vol. 22, no. 8, pp. 4485–4499, Apr. 2022.

[12] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 773–787, Jun. 2009.

[13] M. Wu, F. Ludwig, M. Woltering, D. Wuebben, A. Dekorsy, and S. Paul, "Analysis and implementation for physical-layer network coding with carrier frequency offset," in *Proc. Int. ITG WSA*, Mar. 2014, pp. 1–8.

[14] D. Wübben and Y. Lang, "Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems," in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.

[15] L. You, S. C. Liew, and L. Lu, "Network-coded multiple access II: Toward real-time operation with improved performance," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 264–280, Feb. 2015.

[16] L. Lu, L. You, Q. Yang, T. Wang, M. Zhang, S. Zhang, and S. C. Liew, "Real-time implementation of physical-layer network coding," in *Proc. 2nd Workshop Softw. Radio Implementation Forum*, Aug. 2013, pp. 71–76.

[17] J. Park, D. J. Ji, and D.-H. Cho, "High-order modulation based on deep neural network for physical-layer network coding," *IEEE Wireless Commun. Lett.*, vol. 10, no. 6, pp. 1173–1177, Feb. 2021.

[18] X. Wang and L. Lu, "Implementation of DNN-based physical-layer network coding," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7380–7394, Jan. 2023.

[19] Y. Wang, M. Chen, T. Luo, W. Saad, D. Niyato, H. V. Poor, and S. Cui, "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2598–2613, Jul. 2022.

[20] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2300–2316, Jun. 2022.

[21] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep learning enabled semantic communications with speech recognition and synthesis," *IEEE Trans. Wireless Commun.*, pp. 1–1, Feb. 2023.

[22] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic communications for video conferencing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 230–244, Jan. 2023.

[23] G. Zhang, Q. Hu, Z. Qin, Y. Cai, G. Yu, X. Tao, and G. Y. Li, "A unified multi-task semantic communication system for multimodal data," Aug. 2022. [Online]. Available: https://arxiv.org/abs/2209.07689

[24] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2023.

[25] S. F. Yilmaz, C. Karamanli, and D. Gündüz, "Distributed deep joint source-channel coding over a multiple access channel," Mar. 2023. [Online]. Available: https://arxiv.org/abs/2211.09920

[26] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Jul. 2022.

[27] X. Luo, B. Yin, Z. Chen, B. Xia, and J. Wang, "Autoencoder-based semantic communication systems with relay channels," in *Proc. IEEE ICC Workshops*, Jul. 2022, pp. 711–716.

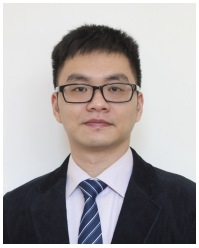
[28] C. Bian, Y. Shao, H. Wu, and D. Gündüz, "Deep joint source-channel coding over cooperative relay networks," Nov. 2022. [Online]. Available: https://arxiv.org/abs/2211.06705

[29] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Nov. 2021.

[30] W. Yang, Y. Cai, J. Hu, and W. Yang, "Channel estimation for two-way relay OFDM networks," *EURASIP J. Wireless Commun. Netw.*, 2010.

[31] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless image transmission using deep source channel coding with attention modules," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2315–2328, May 2022.

[32] D. B. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, Apr. 2020.
[33] Y. LeCun, C. Cortes, and C. J. C. Burges, "THE MNIST DATABASE of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist
[34] M. Abadi, A. Agarwal, P. Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980
[36] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," Jun. 2021. [Online]. Available: https://arxiv.org/abs/2010.11929



Haoyuan Pan (Member, IEEE) received the B.E. and Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014 and 2018, respectively.

He was a Post-Doctoral Fellow with the Department of Information Engineering, CUHK, from 2018 to 2020. He is currently an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include wireless communications and networks, Internet of Things (IoT), semantic communications, and age of information (AoI).

Shuai Yang received the B.S. degree from Hainan Normal University, Haikou, China, in 2021. He is currently working toward the M.E. degree in Shenzhen University, Shenzhen, China. His research interests include semantic communication and physical-layer network coding (PNC).

Tse-Tin Chan (Member, IEEE) received his B.Eng. (First Class Hons.) and Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong SAR, China, in 2014 and 2020, respectively.

He is currently an Assistant Professor with the Department of Mathematics and Information Technology, The Education University of Hong Kong (EdUHK), Hong Kong SAR, China. Prior to this, he was an Assistant Professor with the Department of Computer Science, The Hang Seng University of Hong Kong (HSUHK), Hong Kong SAR, China, from 2020 to 2022. His research interests include wireless communications and networking, Internet of Things (IoT), age of information (AoI), and semantic communications.

Zhaorui Wang received the Ph.D. degree from The Chinese University of Hong Kong (CUHK) in 2019, and the B.S. degree from University of Electronic Science and Technology of China (UESTC) in 2015. He is now a Research Assistant Professor at the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. He was a Postdoctoral Research Associate at The Hong Kong Polytechnic University from 2019 to 2020, and a Postdoctoral Research Associate at CUHK from 2021 to 2022. He is a recipient of the Hong Kong PhD Fellowship from 2015 to 2018. He has been selected in the post of "Pengcheng Peacock Plan" (Type C) since 2022. His research interests include system design on massive machine-type communications (mMTC), and semantic communications.

Victor C. M. Leung (Life Fellow, IEEE) is a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and Director of the Laboratory for Wireless Networks and Mobile Systems at the University of British Columbia (UBC), Canada. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. Dr. Leung is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, IEEE Network, and several other journals. He received the 1977 APEBC Gold Medal, 1977-1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Fellow of IEEE, the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers".

Jianqiang Li (Member, IEEE) received the B.S. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively. He is a professor with the College of Computer and Software Engineering, Shenzhen University, Shenzhen, China. He is the recipient of the National Science Fund for Distinguished Young Scholars. He led five projects of the National Natural Science Foundation and three projects of the Natural Science Foundation of Guangdong Province, China. His major research interests include robotics, hybrid systems, Internet of Things, and embedded systems.