photonics

MDPI

Article

Noise-Correlated Neural Network Channel Selection for Signal-to-Noise Ratio Enhancement in Holographic Data Storage

Junqian Deng, Dakui Lin, Xiao Lin and Xiaodi Tan *

Key Laboratory of Opto-Electronic Science and Technology for Medicine of Ministry of Education, Fujian Provincial Key Laboratory of Photonics Technology, College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350117, China; qsx20230786@student.fjnu.edu.cn (J.D.); lindk@fjnu.edu.cn (D.L.); xiaolin@fjnu.edu.cn (X.L.)

* Correspondence: xtan@fjnu.edu.cn

Abstract

Neural networks significantly outperform traditional methods in both decoding amplitude-, phase-, and polarization-encoded data pages and suppressing noise within them. However, the mechanism behind neural networks' denoising capability remains not fully understood. We discover that zeroing channels can improve the reconstruction effect of the model. Consequently, this paper presents a method to locate the noise feature objectively from γ, the weights of the Batch Normalization (BN) layer. γ stands for the importance of the channel in the model and $\gamma < 1$ means the channel may contain noise feature. Through experiments, removing the channels that contained a higher proportion of these noisy features, the reconstructed data pages showed a ~2% improvement in Peak Signal-to-Noise Ratio (PSNR) compared to results obtained by directly outputting data without removing the noisy channels. It indicates that neural networks achieve efficient denoising of encoded data pages by adjusting the weight parameters of BN layers, thereby suppressing or enhancing specific channels.

Keywords: neural network; holographic storage; phase encode; auto-encoder

1. Introduction

As one of the next-generation storage technologies, holographic storage technology boasts an extremely high theoretical storage density and ultra-fast data conversion rate [1]. Holographic storage can utilize amplitude [2], phase [3], and polarization [4,5] to achieve one-dimensional or multi-dimensional modulation. Meanwhile, various multiplexing technologies [6–8] can further enhance the storage density of holographic storage.

The most commonly used phase encoding is four-grayscale phase encoding. The difference between amplitude encoding and phase encoding is the code efficiency. Four-level phase encoding has two times more code efficiency than amplitude encoding, because one pixel can represent four states, but in amplitude encoding, one pixel only represents two states. The data page will be uploaded by Spatial Light Modulator (SLM), then after recording and reading, the obtained data page will be decoded. Decoding the phase data page with the traditional iteration method [9] has a problem in retrieval accuracy and speed. The application of deep learning has largely solved these problems.

Deep learning, a type of machine learning algorithm [10], is widely applied in numerous fields [11,12]. Hao et al. [13] realize the prediction of the unknown complex amplitude data page directly from the intensity image without any iterations by training an end-to-end

Check for updates

Received: 6 January 2026
Revised: 26 January 2026
Accepted: 27 January 2026
Published: 29 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

convolutional neural network between the phase-encoded data pages and the corresponding near-field diffraction intensity images. In additional, Hao et al. [14] propose a method that can denoise the captured single-shot image by training a neural network on the relationship between the captured intensity images and the simulation truth images. Tatsuki and Fujimura [15] proposed a polarization and phase modulation scheme and a relative detection method for holographic storage, which can detect the phase and polarization state with a single shot and only one light beam. Furthermore, they introduced machine learning and increased the recording density by 2.37 compared to that without machine learning. Chen R. et al. [16] propose a phase retrieval method with embedded data based on deep learning. Through the strong constraints of embedded data, the amount of training data is reduced to 1/50th of the original, and the bit error rate decreases by 9 times. These studies demonstrate that neural networks possess excellent phase decoding capabilities and denoising capabilities. Even though the neural network has great performance, the decoding of the holographic storage data page is still beset by noise. Many researchers have conducted a lot of great work in addressing the noise problem in holographic storage. Lin Y. et al. [17] show an anti-noise performance analysis based on an end-to-end convolutional neural network, which can effectively correct system imaging aberrations, detector light intensity response, holographic storage medium response non-uniformity, and defocusing noise from the recording objective lens. Chen Y. et al. [18] propose a noise reduction method based on an improved block matching and 3D filtering (BM3D) convolutional neural network. They compare the reconstruction result of BM3D with the denoising convolutional neural network (DnCNN), convolutional blind denoising network (CBDnet), and Wiener filtering. The result shows that BM3D has the advantages of traditional algorithms and deep learning, significantly enhances the robustness of the system, and optimizes the denoising performance. Lin Z. et al. [19] introduce a dilated residual attention and multi-scale convolution U-Net (DRAMCU-Net), which enhances the Signal-to-Noise Ratio (SNR) of holographic data storage images by approximately 30% compared to the traditional U-Net. The above research indicates that noise is still a problem in holographic storage, despite the strong noise processing capability of neural networks. Currently, noise cannot be separated from the data page. The majority of studies in the field of holographic storage treat neural networks as "black boxes" and thus seek to replace the original network architectures with alternatives that exhibit superior noise-handling capabilities.

Actually, we can do plenty of things during the training process, e.g., transfer [20], channel attention [21–23], network slimming [24], etc. The basic auto-encoder is a fully connected network [25,26], designed to address the vanishing gradient problem and exploring gradient problem. After years of development, the auto-encoder has many variants that are suited to different tasks in different fields, e.g., convolution auto-encoder [27], variational auto-encoder [28]. In neural networks, different channels may contain different features. It manifests that the surface layer in a neural network extracts the edge and texture features and is sensitive to noise [29]. This paper, based on the idea of network slimming, proposes a method to locate and eliminate the noise in the neural network. We use a convolution auto-encoder, which is more applicable to image-processing tasks (e.g., feature extracting and image denoising), to leverage the characteristics of the auto-encoder, analyzing how the noise data page denoising function is achieved via the neural network. Through the parameter γ, the weight of the Batch Normalization layer in the network, we can locate the noise feature hidden in the channels, and by removing the noise channels, improve the PSNR of the output by approximately 2%.

2. Neural Network

2.1. The Structure of Neural Network

Auto-encoders are a type of neural network trained in an unsupervised manner to learn the compressed features of raw data [30]. We the choose convolutional auto-encoder (CAE) as our neural network. The CAE changes the fully connected layer into the convolution layer in order to enhance the image-processing ability, but the concrete structure of the CAE is relatively flexible. A CAE consists of three components: encoder, bottle, and decoder. Generally, the encoder and decoder will consist of three convolution layers (or transposed convolution layers). The encoder takes a single-channel input and ultimately outputs 128 channels; the input image, with a size of 128 × 128 pixels, is processed to produce 128 feature maps, each of 32 × 32 pixels, and then the decoder performs the inverse operation of the encoder. Most modern convolutional neural network (CNN) models adopt the Batch Normalization (BN) layer as a standardization method to achieve rapid model convergence and improve model generalization [31]. The grayscale values of the input data page range from 0 to 255, with no negative values, and Rectified Linear Unit (ReLU) activation has a better performance in image-processing tasks [32]. Therefore, ReLU activation is elected as the activation function. At the end of the neural network, Sigmoid activation is used to extract the result given by the neural network, preventing the reconstruction from exceeding the range of [0,1] [33]. Figure 1 presents the architecture of the CAE model; the bottle will contain 128 channels, which will be the key parameter in the experiment later.

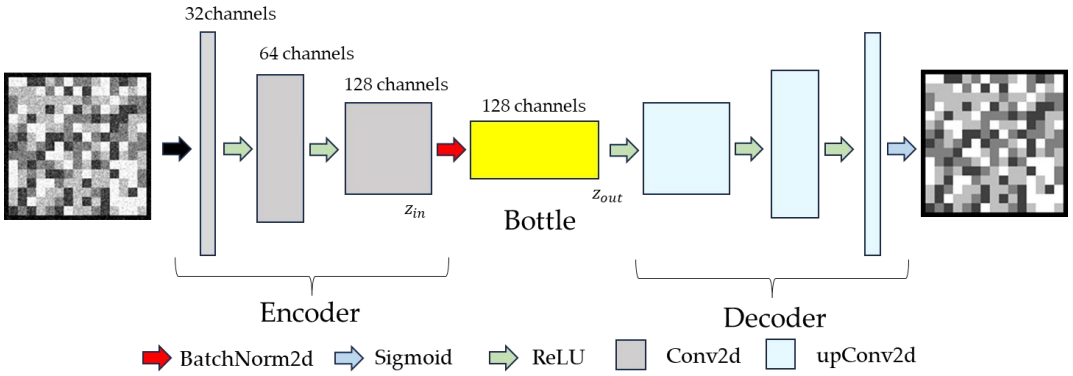


Figure 1. The architecture of CAE. The black arrow means the data input.

The main object processed by the network is images. Therefore, the mean squared error (MSE) function is selected as the evaluation function. MSE can be calculated as follow:

$$MSE = \frac{1}{IWH} \sum_{i=1}^{I} \sum_{u=1}^{W} \sum_{v=1}^{H} (\hat{X}_i(u,v) - X_i(u,v))^2, \quad (1)$$

Here, $\hat{X}_i(u,v)$ represents the output value reconstructed by the model, and $X_i(u,v)$ represents the input value of the model; W and H are the width and height of the data page; I is the mini-batch size in the stochastic gradient descent (SGD) method [34]. Each mini-batch will have an MSE value, and the goal of the evaluation function is to find the minimum MSE between all these mini-batches. The BN layer normalizes the internal activation values based on the mean μ_B and variance σ_B of mini-batch B. Let the input from the previous layer be z_{in} and the output of the BN layer be z_{out}. The network continuously adjusts μ_B,

σ_B through forward propagation and adjusts γ, β through backward propagation. The calculation of the BN layer is as follows:

$$\hat{z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{2}$$

$$z_{out} = \gamma \hat{z} + \beta, \tag{3}$$

γ and β are two parameters generated during the processing of the BN layer. γ is the weights of BN layer, and β is the bias of the BN layer. γ stands for the importance of channels, while β indicates the bias between the channels with the truth value. These parameters are trainable affine transformation parameters in the neural network, which also enable the normalized activation values to be linearly transformed back to any scale. Initially, β is usually 0. The initial value of γ is usually 1, and after training, there will be channels where γ is greater than 1 or less than 1. This means that channels with $\gamma > 1$ have their weights increased, and channels with $\gamma < 1$ have their weights decreased. Therefore, we suppose that among the numerous channels obtained by the neural network when processing noisy data pages, channels with $\gamma < 1$ indicate that the channel contains more noise features and can be eliminated; channels with $\gamma > 1$ represent more features of the original data pages and need to be retained.

2.2. Dataset and Model Training

In this section, we will set up a dataset for training and testing and introduce why we select the Gaussian noise and how we add and use the data page with Gaussian noise. Due to the fact that thermal noise from electronic devices follows a Gaussian noise pattern [35,36] is very common in holographic storage, we choose Gaussian noise as the experiment target. The most common electronic device in holographic storage systems is the SLM, which is responsible for uploading the data page to the optical path. The data page used in the network training is shown in Figure 2a. Figure 2c shows both of them as grayscale images with a specification of 128 × 128 pixels, totaling 4439 data pages in the dataset. The data page using an 8 × 8 macro-pixel stands for a real pixel; the actual size of the data is 15 × 15 pixels. We oversample the pixel to decrease the crosstalk noise when the data page is uploaded to the SLM. In addition, since we do not have a demand for increasing the storage capacity of a single data page, an 8 × 8 oversampling size is adopted. Figure 2b is four different noise levels in the test dataset, which uses a noise-free data page that does not overlap with the training dataset and simulates 15 sets of data with Gaussian noise. The probability density function of Gaussian noise is as follows:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{4}$$

μ is the signal, so when x = μ, there is a raw data page; σ is standard deviation, which stands for the distribution of Gaussian noise. σ represents the amplitude of the signal, so a smaller σ indicates less noise. We set σ as the noise coefficient; the larger the value of σ, the stronger the noise added on the data page [37]. The noise set in Figure 2b ranges from 1 to 70, and all the noise factors are multiples of 5 except for the first one. In order to train the capability of retrieving the original data, the training dataset used in Figure 2a is noise-free. In this paper, we only discuss the Gaussian group because the experiment and the analysis are the same.

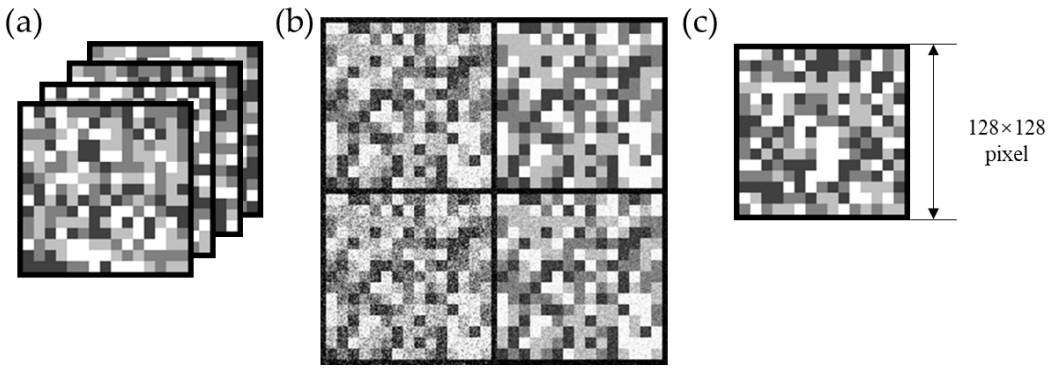


Figure 2. (**a**) Training dataset; (**b**) test dataset with different noise levels; (**c**) the size of data page.

The model was trained on an NVIDIA GeForce RTX 4060 Laptop GPU and Intel® Core™ i9-14900HX for 30 epochs with the training dataset in Figure 2a. The following experiments are conducted on the model obtained in the final round of training. Because of the noise-free training dataset, the trained scale factor will not connect with the noise feature. So, we will not freeze γ in the test; rather, we will allow the back-propagation to update γ with noise features from the test data page. In one test, we only use one data page, and the learning rate is 0.0001. The first update step prohibits convolutional layers from updating in the code and then uses three times the back-propagation to fine-tune γ. The γ change during this process is very small when we compare the model before and after, but the fine-tuning can easily be seen in the result. Every data page in the test dataset will independently correlate themselves with a γ value.

3. Experiments and Results

The results from the model will be compared with the original data page by Peak Signal-to-Noise Ratio (PSNR). PSNR can be calculated as follows:

$$PSNR = 10log_{10}(\frac{MAX_I^2}{MSE}) \text{ (dB)}, \quad (5)$$

MAX is the max gray-scale of the data page, which is 255 in our data page. MSE is the mean squared error function mentioned in Section 2.1 Equation (1).

3.1. Channel Classification and Validation

In this section, we will focus on the feasibility verification. The neural network will generate 128 channels in total at the bottle, and we will randomly zero 15 of them. Different channels may contain different features of the data page, but most channels contain useful features. So, we only focus on zeroing out some channels and retaining most of them in order to retain the original features of the data page to the greatest extent possible. Firstly, we use a noise level of $\sigma = 30$ in the test and randomly zero a given number of channels, ranging from 10 to 30, to test whether zeroing channels affects the neural network output. In total, we randomly zero five times. Figure 3 shows the average and the standard deviation of the PSNR. Generally, the PSNR value is positively correlated with the data page reconstruction quality. A PSNR exceeding 30 dB is regarded as having no adverse effect. But actually, around 30 dB might also be accepted after evaluation against other criteria like BER (Bit Error Rate) [38]. According to the result in Figure 3, zeroing 10, 15, 20, and 25 channels can achieve better reconstruction than no modification, and zeroing 15 channels is the best. Therefore, we take the 15 channels group, which allows more room for adjustment in the following experiment.

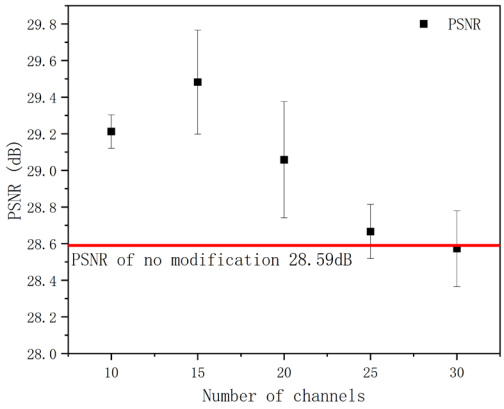


Figure 3. Result of randomly zeroing 10–30 channels; above the redline means a better reconstruction than no modification.

We firstly test the reconstruction result of 15 levels of noise without modifying channels. Figure 4 shows the results of the no modification group. Except for σ = 1, the PSNR is always better than the raw data page, which indicates that the model possesses a certain denoising capability, even without modifying the channels. When σ = 1, the no modification group is worse than the raw data page. We speculate that σ = 1 is a very low noise level, and this noise impact on the data page is negligible, so the reconstruction makes no effort to improve the PSNR. Secondly, we randomly zero out 15 channels to test whether the reconstruction result will be better after zeroing out channels. We speculate that noise intensity will influence the noise features in channels. Therefore, we test with 15 different levels of noise in order to increase noise features in channels. In Figure 4, when σ = 25, the PSNR of the modified group exceeds the non-modification group. Prior to σ = 20 (including 20), the result indicates the contrary. We conjecture that when the noise level is low, noise has little impact on the reconstruction, but beyond σ = 25, noise may have more influence on the network, and zeroing channels may diminish their influence. When we continually increase the noise, zeroing channels will have an advantage period, and the limitation is around σ = 70. Therefore, we conclude that the method of zeroing out channels is workable, and some of the randomly zeroed channels are covered by noise features when the input noise reaches a certain level.

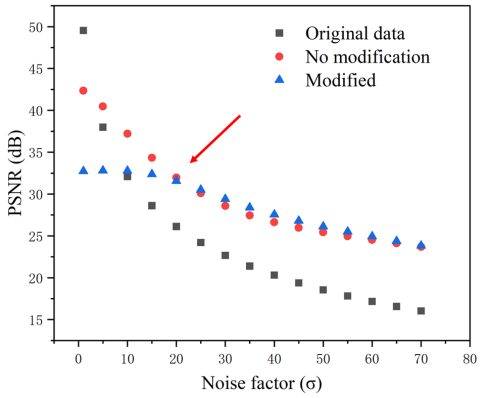


Figure 4. Original data are the PSNRs of raw data; no modification is the reconstruction result of the network without zeroing channels; modified is the result of randomly zeroing 15 channels. Red arrow indicates that the result of zeroing out channels is close to the result of non-modified group here, and then overtakes later.

3.2. Search Noise Feature Manually

The purpose of this part is to locate noise-containing channels, designate 15 channels deemed to contain noise for zeroing out, and verify whether the results in Figure 5 can still be reproduced. The 128 output channels of the encoder can be roughly categorized into eight types, as shown in Figure 5. In order to improve the distinguishability of different phases, the feature is colored by the 'viridis' color map. The classification is according to pattern characteristics of the feature maps, but it is a rough classification that allows us to put similar features together. For example, Figure 5a has a prominent feature with an iconic pattern at the bottom right corner, which is inverse in Figure 5c. Though we have classified the features, we will focus more on the subsequent testing methods. So, we can classify these features as different types. However, these eight types only cover approximately 90% of all the features. Figure 5 only shows the most representative categories. Then, each of these eight groups and those not included in the classification are further subdivided into several subgroups with five channels per subgroup (if the last subgroup is fewer than five, then previous channels can be used to fill to five).

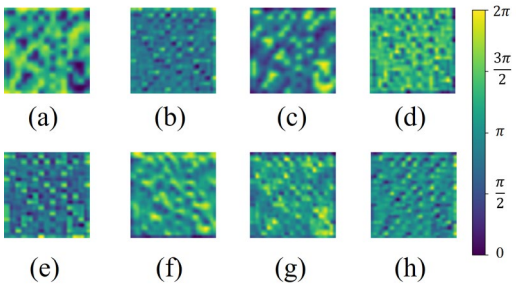


Figure 5. Eight different categories of features and the color map. (**a**–**h**) show difference between the eight types of features.

Subsequently, we will randomly zero out five channels and calculate the PSNR of the result as a standard to make a horizontal comparison with those subgroups. Those subgroups' PSNRs that are beyond the standard group will be selected. Finally, 15 channels are screened out, and the serial numbers of these channels are shown in Table 1. Among these, 15 channels are chosen to repeat the experiment under the same conditions as in Section 3.1, except the channels are selected instead of choosing at random. Figure 6 shows that after zeroing out 15 elected channels, we obtain a similar result to Figure 5, in which the data trends and the PSNR turning point of non-modification and modification are the same. The elected channels served the same function as the random channels and proved that the noise features hidden in the channels can be located. γ presented in Table 1 is the weight of the BN layer that will be the evaluation criterion in Section 3.3. We boldface the channels with $\gamma < 1$, which will be discussed in the next section.

3.3. Introduce the Scale Factor γ

The previous experiments indicated that the neural network extracts the noise features from the noisy data pages during the feature extraction process. To verify whether these channels contain noise features, this part is designed to confirm the aforementioned speculation. In the neural network model, the encoder includes a Batch Normalization (BN) layer at the end (the red arrow in Figure 1), and the scaling factors γ (weights of Batch Normalization layer) can represent the importance of each channel in the network. A higher weight parameter of a channel indicates a greater importance of that channel in the network. Table 2 lists the 15 lowest channels of the total 24 channels with $\gamma < 1$, and half of the channels are the same as those used in Section 3.2. It can be observed that the weights of these 15 channels do not differ significantly from 1, and their enhancement or suppression

effects on the channels are not very pronounced. This phenomenon is speculated for two reasons: first, although some data pages are added with significant noise, the noise is of a single type, unlike the mixed multiple noise patterns in real materials; second, although the addition of noise affects the reconstruction performance, the difference between noisy data pages and noise-free (clean) data pages is not substantial, so no large γ values are required for correction. Figure 7a shows the result of zeroing 15 channels with $\gamma < 1$, while Figure 7b is the comparison of zeroing 15 $\gamma < 1$ channels and zeroing 15 elected channels. Figure 7b shows that zeroing out 15 channels with $\gamma < 1$ has a better performance as the noise is added, and both of them reached a turning point around $\sigma = 20$. At the medium noise level (σ ranging from 25 to 45), zeroing out channels with $\gamma < 1$ gains an approximately 0.5 dB advantage compared with zeroing elected channels. Across all three experiments, the noise-free groups exhibited degraded reconstruction quality due to channel zeroing, while the noisy groups showed improved reconstruction quality when the noise reached a certain level. Therefore, it is concluded that during the neural network-based reconstruction of data pages, some channels extract noise features, which degrade the network's reconstruction quality. These noise-related features can be screened out through the weight γ of the BN layer.

Table 1. The elected channels.

Channel number	**1** *	3	**6**	**28**	**30**
γ	**0.9836**	1.0071	**0.9985**	**0.9943**	**0.9944**
Channel number	31	**34**	45	48	**110**
γ	1.0130	**0.9941**	1.0007	1.0168	**0.9993**
Channel number	112	120	**122**	123	124
γ	1.0112	1.0007	**0.9958**	1.0137	1.0019

* The boldface channels are $\gamma < 1$.

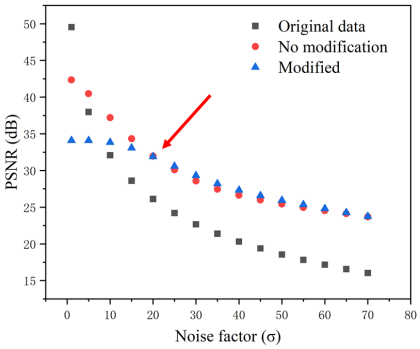


Figure 6. Original data is the PSNR of raw data; No modification is reconstruction result of the network without zeroing channels; Modified is the result zeroing 15 channels that filtering from 128 channels. Red arrow indicates that the result of zeroing out channels is close to the result of non-modified group here, and then overtakes later.

Table 2. Channels with $\gamma < 1$.

Channel number	**1** *	**6**	10	12	15
γ	**0.9836**	**0.9985**	0.9811	0.9952	0.9803
Channel number	**28**	**30**	**34**	38	81
γ	**0.9943**	**0.9944**	**0.9941**	0.9918	0.9918
Channel number	84	106	107	**110**	**122**
γ	0.9868	0.9954	0.9781	**0.9993**	**0.9958**

* The boldface channels are those existing in Table 1.

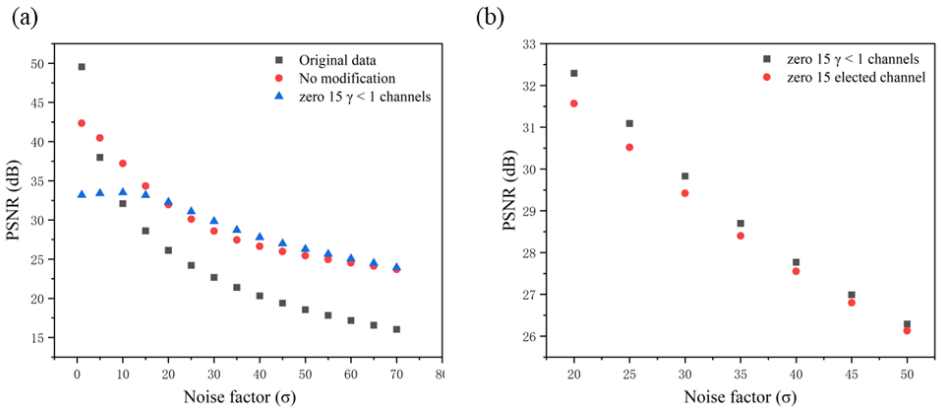


Figure 7. (**a**) Original data are the PSNRs of raw data; no modification is reconstruction result of the network without zeroing channels; modified is zeroing out 15 channels with $\gamma < 1$. (**b**) Result with σ range in a closed interval [20,50] to compare the result of zeroing 15 $\gamma < 1$ channels and zeroing 15 elected channels.

4. Conclusions

Aiming at the noise problem in holographic storage phase retrieval, we investigated how the neural network achieves noise elimination during data page decoding using a CAE neural network and discovered the relationship between the channel and the noise. This work proposes a method in which γ, the weight of the BN layer, can help locate the noise features in the neural network's channel, and eliminating the channel highly correlated to noise can improve the PSNR of reconstruction. We validate that zeroing out channels can improve the performance of neural network, and zeroing 15 channels has the best performance. Via zeroing out channels, we decrease the number of channels strongly correlated with noise, obtaining an approximately 2% enhanced PSNR compared to the result of no modification. After connecting with γ, we locate the noise feature in a more objective way and gain a 0.5 dB base improvement when zeroing out manually elected channels. All experimental results indicate that some of the zeroed channels harbor noise-related features, and we can remove these channels to enhance the reconstruction performance. Additionally, the improvement will reach the maximum when the σ of Gaussian noise is between 20 and 50.

The network used in this work and the simulated noise on data pages are relatively simple. Neural network models applied to holographic storage in practical scenarios and noise patterns on data pages requiring decoding are more complex, and roughly zeroing out channels cannot completely eliminate the impact of these channels on the entire neural network model. In subsequent research, a more scientific method is needed to manipulate the channels in the model. The objectives of future work include verifying the simulation-based experimental results using real captured data pages on deeper neural networks and equipping the model with a function to automatically eliminate noise-containing channels.

Author Contributions: Conceptualization, J.D.; methodology, J.D.; validation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, J.D., D.L. and X.T.; supervision, X.L., D.L. and X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, the grant number is U22A2080.

Data Availability Statement: The data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Van Heerden, P.J. Theory of optical information storage in solids. *Appl. Opt.* **1963**, *2*, 393–400. [CrossRef]
2. Brian, M.; Mark, A. Sparse modulation coding for increased capacity in volume holographic storage. *Appl. Opt.* **2000**, *39*, 6681–6688. [CrossRef] [PubMed]
3. Lin, X.; Hao, J.; Zheng, M.; Dai, T.; Li, H.; Ren, Y. Optical holographic data storage—The time for new development. *Opto-Electron. Eng.* **2019**, *46*, 180642-1–180642-15. [CrossRef]
4. Kuroda, K.; Matsuhashi, Y.; Fujimura, R.; Tsutomu, S. Theory of polarization holography. *Opt. Rev.* **2011**, *18*, 374–382. [CrossRef]
5. Wang, J.; Tan, X.; Qi, P.; Wu, C.; Huang, L.; Xu, X.; Huang, Z.; Zhu, L.; Zhang, Y.; Lin, X.; et al. Linear polarization holography. *Opto-Electron. Sci.* **2022**, *1*, 210009-1–210009-20. [CrossRef]
6. Yang, K.; Liu, F.; Liang, S.; Xiang, M.; Han, P.; Liu, J.; Dong, X.; Wei, Y.; Wang, B.; Shimizu, K.; et al. Data-driven polarimetric imaging: A review. *Opto-Electron. Sci.* **2024**, *3*, 230042-1–230042-44. [CrossRef]
7. Zheng, S.; Tan, J.; Liu, H.; Lin, X.; Saita, Y.; Nomura, T.; Tan, X. Orthogonal matrix of polarization combinations: Concept and application to multichannel holographic recording. *Opto-Electron. Adv.* **2024**, *7*, 230180-1–230180-10. [CrossRef]
8. Li, K.; Wang, Y.; Pi, D.; Li, B.; Luan, H.; Fang, X.; Chen, P.; Lu, Y.; Gu, M. Multi-dimensional multiplexing optical secret sharing framework with cascaded liquid crystal holograms. *Opto-Electron. Adv.* **2024**, *7*, 230121-1–230121-8. [CrossRef]
9. Lin, X.; Huang, Y.; Shimura, T.; Fujimura, R.; Tanaka, Y.; Endo, M.; Nishimoto, H.; Liu, J.; Li, Y.; Liu, Y.; et al. Fast non-interferometric iterative phase retrieval for holographic data storage. *Opt. Express* **2017**, *25*, 30905–30915. [CrossRef]
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
11. Wu, S.; Yang, Z.; Ma, C.; Zhang, X.; Mi, C.; Zhou, J.; Guo, Z.; Jin, D. Deep learning enhanced NIR-II volumetric imaging of whole mice vasculature. *Opto-Electron. Adv.* **2023**, *6*, 220105. [CrossRef]
12. Li, Z.; Sun, J.; Fan, Y.; Jin, Y.; Shen, Q.; Trusiak, M.; Cywińska, M.; Gao, P.; Chen, Q.; Zuo, C. Deep learning assisted variational Hilbert quantitative phase imaging. *Opto-Electron. Sci.* **2025**, *2*, 220023-1–220023-11. [CrossRef]
13. Hao, J.; Lin, X.; Lin, Y.; Chen, M.; Chen, R.; Situ, G.; Hideyoshi, H.; Tan, X. Lensless complex amplitude demodulation based on deep learning in holographic data storage. *Opto-Electron. Adv.* **2023**, *6*, 220157. [CrossRef]
14. Hao, J.; Lin, X.; Chen, R.; Lin, Y.; Liu, H.; Song, H.; Lin, D.; Tan, X. Phase retrieval combined with the deep learning denoising method in holographic data storage. *Opt. Contin.* **2022**, *1*, 51–62. [CrossRef]
15. Tatsuki, T.; Fujimura, R. Deep-learning-enhanced single-shot detection of polarization-and phase-encoded signals for holographic data storage. ODS 2025: Industrial Optical Devices and Systems. *SPIE* **2025**, *13610*, 1361002. [CrossRef]
16. Chen, R.; Wang, J.; Zhang, S.; Lin, D.; Li, X.; Zheng, J.; Cao, Q.; Hao, J.; Lin, X.; Tan, X. Deep learning-based phase retrieval with embedded data for holographic data storage. *Opt. Express* **2025**, *33*, 12731–12746. [CrossRef]
17. Lin, Y.; Ke, S.; Song, H.; Liu, H.; Yang, R.; Lin, D.; Li, X.; Zheng, J.; Cao, Q.; Hao, J.; et al. Anti-noise performance analysis in amplitude-modulated collinear holographic data storage using deep learning. *Opt. Express* **2024**, *32*, 29666–29677. [CrossRef]
18. Chen, Y.; Fan, Y.; Zhang, G.; Wang, Q.; Li, S.; Wang, Z.; Dong, M. Speckle noise suppression of a reconstructed image in digital holography based on the BM3D improved convolutional neural network. *Appl. Opt.* **2024**, *63*, 6000–6011. [CrossRef]
19. Lin, Z.; Hao, J.; Lin, Y.; Liu, H.; He, C.; Lin, D.; Lin, X.; Tan, X. SNR enhancement for low-SNR amplitude-modulated holographic data storage based on deep learning. *Opt. Express* **2025**, *33*, 49488–49503. [CrossRef]
20. Neyshabur, B.; Hanie, S.; Zhang, C. What is being transferred in transfer learning? *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 512–523.
21. Jetley, S.; Nicholas, A.; Namhoon, L.; Philip, H.S. Learn to pay attention. *arXiv* **2018**, arXiv:1804.02391. [CrossRef] [PubMed]
22. Schlemper, J.; Oktay, O.; Chen, L.; Matthew, J.; Knight, C.; Kainz, B.; Glocker, B.; Rueckert, D. Attention-gated networks for improving ultrasound scan plane detection. *arXiv* **2018**, arXiv:1804.05338. [CrossRef]
23. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999. [CrossRef]
24. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
25. Bourlard, H.; Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.* **1988**, *59*, 291–294. [CrossRef]
26. Hinton, G.E.; Zemel, R.S. Autoencoders, minimum description length, and Helmholtz free energy. *Adv. Neural Inf. Process. Syst.* **1994**, *6*, 3–10.
27. Zhang, Y. A Better Autoencoder for Image: Convolutional Autoencoder. ICONIP17-DCEC. 2018. Available online: https://users.cecs.anu.edu.au/ Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf (accessed on 26 January 2026).
28. Le, Q.; Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Ng, A. On optimization methods for deep learning. In Proceedings of the 28th International Conference on International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 265–272.
29. Zeiler, M.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2014; pp. 818–833. [CrossRef]

30. Pierre, B. Autoencoders, unsupervised learning and deep architectures. *Workshop Conf. Proc.* **2012**, *27*, 37–49.
31. Ioffe, S.; Christian, S. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. Mach. Learn. PMLR* **2015**, *37*, 448–456.
32. Maninis, K.K.; Pont-Tuset, J.; Arbeláez, P.; Van Gool, L. Deep retinal image understanding. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer International Publishing: Cham, Switzerland, 2016; pp. 140–148. [CrossRef]
33. Xu, X.; Wang, X.; Luo, W.; Wang, H.; Sun, Y. Efficient Computer-Generated Holography Based on Mixed Linear Convolutional Neural Networks. *Appl. Sci.* **2022**, *12*, 4177. [CrossRef]
34. Ferguson, T.S. An inconsistent maximum likelihood estimate. *J. Am. Stat. Assoc.* **1982**, *77*, 831–834. [CrossRef]
35. Gu, C.; Sornat, G.; Hong, J. Bit-error rate and statistics of complex amplitude noise in holographic data storage. *Opt. Lett.* **1996**, *21*, 1070–1072. [CrossRef]
36. Gu, C.; Dai, F.; Hong, J. Statistics of both optical and electrical noise in digital volume holographic data storage. *Electron. Lett.* **1996**, *32*, 1400–1402. [CrossRef]
37. Ribeiro, M.I. *Gaussian Probability Density Functions: Properties and Error Characterization*; Institute for Systems and Robotics: Lisboa, Portugal, 2004.
38. Bovik, A.C. *Handbook of Image and Video Processing*; Academic Press: Cambridge, MA, USA, 2010.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.