Facilitating real-time LED-based photoacoustic imaging with DenP2P: An optimized conditional generative adversarial deep learning solution

Cite as: AIP Advances 15, 055227 (2025); doi: 10.1063/5.0259072
Submitted: 21 January 2025 • Accepted: 29 April 2025 •
Published Online: 21 May 2025

Avijit Paul and Srivalleesha Mallidi[a)]

AFFILIATIONS

Biomedical Engineering, Tufts University, Medford, Massachusetts 02155, USA

[a)]Author to whom correspondence should be addressed: srivalleesha.mallidi@tufts.edu. Tel.: +1-617-627-4783

ABSTRACT

Photoacoustic imaging (PAI) benefits from the optical absorption contrast of the tissue while achieving greater depth information with ultrasound resolution than the other optical imaging platforms. The recent advancement of moderate pulse width LED (e.g., Acoustic-X) illuminating devices makes PAI more affordable, mobile, and fast with a trade-off for low illumination energy, leading to low signal-to-noise-ratio (SNR) images, which are averaged over time to get high SNR images. Signal quality can be improved by traditional noise removal algorithms, but deep learning models outperform non-learning methods. Although the most widely used U-Net architecture removes noise, it compromises the structural similarity, introduces blur, and causes edge artifacts. To mitigate those issues, we explored a gamut of architectural options of the Pix2Pix-based conditional generative adversarial network (cGAN) analyzing objective functions, optimizers, activation, and normalization layers. The optimized denoising cGAN model based on Pix2Pix architecture (DenP2P) is tested with a variety of out-of-class biological test data such as *in vivo* mouse tumor, *ex vivo* kidney, heart, and liver spatially different from training examples. The network is also tested for noise distribution type invariancy concerning Gaussian white, Poisson, speckle, and salt and pepper noise. The frequency domain's magnitude spectrum analysis explains less blurring for the generated outcomes compared than U-Net. In addition, the persistence homology diagrams (birth, death, and lifetime) underscored DenP2P's efficacy in diminishing noisy topological features, fostering the emergence of stable and resilient structures. Overall, the optimized DenP2P model generates high SNR images with appreciable peak SNR and structural similarity index.

I. INTRODUCTION

Accurate clinical diagnosis plays a crucial role in effective medical treatment as it serves as the initial step in the workflow. Consequently, biomedical imaging has emerged as a key research area. Photoacoustic (PA) imaging (PAI), an imaging modality conceptualized over 150 years ago,[1] became a reality with the advent of laser technology. This hybrid imaging technique[2–4] traditionally combines high optical contrast through light illumination in the NIR wavelengths with ultrasound reception, which offers significant depth resolution with 3–4 orders less attenuation. Laser systems, with their small pulse width (around 10–100 ns), high energy output, and tunability, are well-suited for research spaces and large hospitals. However, their high cost, bulkiness, and lack of portability make them impractical for low resource clinics.[5]

An alternative to laser-based systems is light-emitting detector (LED) based optical systems. These systems offer a more affordable and portable solution with acceptable pulse widths (~100–150 ns). One commonly used LED-based system in PAI is LED Acoustic-X.[6–8] However, a major challenge with LED-based systems is their low energy output in the nanojoule range.[9] This limitation leads to low signal-to-noise ratio (SNR) images when using a small number of frame averages (around 128 frames). To overcome this issue, researchers tend to average a large number of captured frames (~25600 frames) over time due to the high pulse repetition frequency of LED-based systems (ranging from 1 to 4 kHz). Averaging

helps cancel out high-frequency noise and improve SNR but significantly increases the imaging time (0.15 Hz). Consequently, there exists a trade-off between achieving high SNR and real-time imaging.

One approach to address this trade-off is to employ downstream noise removal processes that enhance SNR and bridge the time gap. Traditional noise removal algorithms[10] are not highly effective in reducing noise without prior knowledge of the noise characteristics, which can be stochastic and unknown in real-life situations. Alternatively, machine learning (ML) algorithms offer a promising solution as they can learn from data. However, ML algorithms require feature engineering,[11] which introduces the potential for human errors. The recent advancements[12,13] in graphics processing units (GPUs), which provide parallel processing capabilities, along with the abundance of datasets, have paved the way for state-of-the-art deep learning (DL) architectures to excel in various problem domains. Unlike traditional machine learning algorithms, DL networks do not rely on pre-calculated features, making them highly suitable for a wide range of applications. PAI is no exception and has emerged as a promising field for DL implementations.[14–16] While previous studies have explored DL in PAI, many of them focused on laser-based illumination,[17–21] leading to challenges such as artifacts or data sparsity due to a limited number of detectors in techniques such as PA computed tomography. In addition, some studies[22] employed complex network architectures that require a large amount of training data, while others[23,24] tested their networks using test data that closely resembled the training data in terms of spatial structure. In one of our recent studies utilizing the U-Net architecture,[25] we observed a significant noise reduction; however, high frequencies were affected more severely than the ground truths. This degradation resulted in blurring and introduced edge effects in the Fourier domain, impacting the structural similarity of the reconstructed images. While the peak SNR (PSNR) might not be heavily affected, the overall structural fidelity is compromised.

Combining a U-Net generator and adversarial loss in a conditional generative adversarial network (cGAN[26])—Pix2Pix-based[27] image-to-image translator supervised architecture might lead to better noise invariancy and improved image quality by striking a balance between pixel-wise accuracy and perceptual realism. The U-Net model is trained using only pixel-wise losses, which focus on per-pixel differences between the ground truth and the predicted image. The adversarial loss in the GAN setup introduces a discriminator that learns to distinguish between real and generated images.[28,29] This will force the U-Net generator to produce images that not only minimize pixel-wise differences but also look visually realistic, which might help preserve finer textures and high-frequency details. This additional constraint from the discriminator will improve robustness to noise and mitigates over-smoothing (blurring). In the Pix2Pix framework, the discriminator can ensure that the generated image preserves local textures and patterns across the image, enforcing not just pixel-level accuracy but also local consistency, which will improve overall image quality and metrics such as structural similarity index (SSIM) and PSNR. In another recent study,[30] we introduced a cycle consistent GAN-based approach for denoising LED-based PA images, leveraging an unsupervised framework that does not require paired noisy-clean image datasets. While this method addresses challenges in obtaining precisely matched cross-sectional images with different noise levels, it comes with notable trade-offs. The training time is significantly longer (~4x slower than traditional GAN setups), and the resulting denoised images exhibit lower PSNR and SSIM values compared to our proposed DenP2P approach. In addition, CycleGAN-based denoising may either omit low-intensity signals or introduce artifacts, which are difficult to quantify using standard image quality metrics. In contrast, our supervised learning-based approach not only ensures improved SNR but also preserves low-intensity signals more effectively, as validated by both quantitative and qualitative assessments. Other self-supervised methods, such as Noise2Noise,[31–34] have the advantage of not requiring paired datasets, making them particularly useful when clean ground truth images are unavailable. However, these methods come with notable limitations. Noise2Noise, for instance, assumes that noise is statistically independent across different acquisitions of the same scene, which may not hold true in real-world experimental conditions, especially in PAI, where noise can be highly correlated across frames. Furthermore, self-supervised approaches often struggle with structured noise patterns and may not generalize well to unseen noise distributions. In contrast, our study benefits from the availability of paired training data, making a supervised deep learning model the more suitable choice. Supervised methods can exploit the direct pixel-wise correspondence between noisy and clean images, enabling the model to learn explicit noise-to-signal mappings. This often results in superior denoising performance and higher fidelity reconstructions compared to self-supervised approaches, which rely on indirect learning signals. Therefore, while self-supervised techniques offer clear advantages in scenarios lacking paired data, a supervised approach is the optimal choice when paired data are available. In this current work, we considered a latent variable-based implicit generative model—DenP2P (Denoising Pix2Pix) where the network learns a distribution mapping from the noisy input domain to the paired denoised output space. We took the basic architecture of the Pix2Pix model and tried a gamut of generator loss functions (different combinations of *l1*, *l2*, *Huber*, SSIM, and PSNR loss) and optimization methods (*Adam*, *Nadam*, *rectified Adam*, *AdaBelief*, and *Yogi*) to calculate the PSNR and SSIM. We observed that *Adam* is comparatively the best optimizer with a combination of original GAN, *Huber*, and SSIM generator loss. Even though there is no statistically significant difference among the optimizers with respect to the image quality metrics—PSNR and SSIM, we chose *Adam* as our representative optimizer due to less training time (93 s per 1000 steps) and more robust generalization. Next, we experimented with delta (δ) values for *Huber* loss function and found the optimal value of δ as 1. We also experimented with the activation [Rectified Linear Unit (*ReLU* and *leaky ReLU*)] and normalization layers (*batch*, *weight*, and *spectral*) and selected *ReLU* with *batch* normalization layers. In summary, we built an optimized model (DenP2P) with an *Adam* solver, a combination of original GAN, *Huber* (δ = 1), and SSIM loss as the generator loss, *ReLU* activation, and *batch* normalization layers.

We evaluated the performance of our model using out-of-class *ex vivo* kidney, heart, and liver samples, as well as *in vivo* mouse tumor samples. These samples were spatially and morphologically different from the training data. To assess the network's ability to handle different noise distribution types, we introduced artificially added Gaussian white, Poisson, salt and pepper (S&P), and speckle noise. We compared the performance of the *Adam* and

rectified Adam optimizers for those noise types and found that they yielded similar results in terms of PSNR and SSIM, with no statistically significant difference. Frequency domain analysis of the DenP2P model also revealed no observable blurring, indicating that the reduction of high frequencies and edge effects was effectively managed. Furthermore, we examined the results of DenP2P through a topological lens using homology persistence diagrams following the filtered Vietoris–Rips complexes.[35–37] We compared the birth and death of the topological characteristics identified in both low (LA) and high (HA) frame-averaging scenarios, as well as DenP2P outcomes, up to second homology dimension. While our previous work focused on convolution-based U-Net architectures for image denoising,[25,38] the present study introduces a substantially advanced generative framework—DenP2P—that offers a suite of innovations tailored to the complex noise characteristics of our imaging modality. Unlike standard Pix2Pix implementations,[39–42] DenP2P is extensively optimized through the integration of nine customized loss functions, systematic experimentation with diverse optimizers, and critical architectural enhancements to both the generator and discriminator. These modifications, refined via ablation studies, significantly improve the model's capacity to preserve structural fidelity under varying noise conditions. Notably, we employ frequency magnitude spectrum analysis to quantitatively demonstrate DenP2P's ability to reduce blurring and maintain image sharpness far more effectively than prior methods. In addition, DenP2P exhibits remarkable noise-invariant performance across a spectrum of noise types—including Gaussian, Poisson, S&P, and speckle noise—underscoring its robustness and generalizability. As another layer of analysis, we incorporate persistence diagram-based topological evaluations, offering a unique perspective on the structural integrity of denoised outputs beyond traditional image quality metrics. Collectively, these contributions represent a significant advancement over conventional approaches, not merely in network design but through a holistic optimization strategy that integrates architectural refinement, customized learning objectives, topological insights, and frequency-based sharpness evaluation. Overall, our downstream DenP2P cGAN model demonstrated significant noise reduction without noticeable image blur distortion and exhibited invariance to various noise distribution types.

II. METHODOLOGY

A. Imaging platform

All the images were acquired using Cyberdyne Inc.'s Acoustic-X LED-based PAI system. The linear array transducer utilized had a center frequency of 7 MHz with a −6 dB bandwidth of 80% and consisted of 128 elements. The illumination source employed was LEDs emitting at a wavelength of 850 nm, providing pulses with a duration of 30 ns and a pulse repetition frequency of 4 kHz. The gain settings were kept constant throughout the experiment, ranging from 60 to 67 dB depending on whether *in vitro* or *in vivo* samples were under examination. Different gain settings were employed for *in vitro* and *in vivo* experiments to account for variations in signal attenuation, background noise, and optical fluence distribution. In *in vivo* imaging, we used higher gain (66–67 dB) to compensate for tissue-induced signal loss and optimize contrast, whereas for *in* vitro imaging, slightly lower gain (60–61 dB) due to more uniform optical fluence and reduced background noise was sufficient. For the murine models and metal phantoms, the dynamic range was adjusted to 19 dB and 19–25 dB, respectively. High-frame rate acquisitions were performed at 30 Hz, with data averaging over 128 image frames, while low frame rate acquisitions were conducted at a rate of 0.15 Hz, resulting in images generated from 25 600 frames. The acquired beamformed images were subsequently processed using custom-coded MATLAB (R2021b) for cropping and scaling to a size of 256 × 256 pixels. MATLAB was also employed for implementing image quality metrics analysis.

B. Phantoms

Training data: We used metal frames, wires of different shapes, and graphite rods at two different setups—LA and HA for our training inputs and labels, respectively. The LED Acoustic-X captured images are cropped accordingly and resized to 256 × 256. In total, we collected 800 snapshots of the objects at various spatial positions and depths. In addition, we performed extensive data augmentation, including scaling, shifting, rotating, and stretching, to significantly increase the effective size and variability of the dataset. These augmentations help the model generalize better to unseen samples. The dynamic range varied between 19 and 25 dB, and the gain was set to 60–61 dB.

Test data: Our test data encompassed a diverse range of samples, including in-class metal phantoms and out-of-class *ex vivo* biological organs, as well as *in vivo* tumors in female BALB/c nude mice (8 weeks old when the tumor was implanted). The *ex vivo* organ imaging involved capturing snapshots of the heart, kidney, and liver from mice. For the *in vivo* experiments, subcutaneous injections of AsPC-1 human pancreatic cancer cells suspended in a mixture of Matrigel (BD Biosciences) and phosphate-buffered saline (1:1 v/v) were administered to nude mice. The tumors were allowed to grow to a size of ~300–400 mm^3 within 55–60 days post-inoculation, exhibiting a heterogeneous microenvironment consisting of both vascular and avascular regions.

Before imaging, the mice were anesthetized with 2% isoflurane and positioned on a specially designed platform immerged in a water bath, with their heads elevated above the water level to ensure their safety. The isoflurane concentration was subsequently reduced to 1%–1.5% during the imaging process to maintain anesthesia. A total of 8 mice were included in this investigation, with three frames captured at intervals of 4–5 mm for each mouse. The experimental protocols adhered to the guidelines set by the Institutional Animal Care and Use Committee of Tufts University. For noise distribution type invariancy, we corrupted the ground truth (HA) with the following noise types:

- *Gaussian white noise:* This is a form of additive noise whose probability density function follows a normal distribution with 0.01 variance.[43–46] The PDF can be mathematically expressed as $P(g) = \sqrt{\frac{1}{2\pi\sigma^2} \cdot e^{\frac{-g^2}{2\sigma^2}}}$, where g is the gray value, mean = 0, and σ is the variance.
- *Poisson:* This noise distribution depends on the input data type. PDF for this noise type is $P(N) = \exp(-\langle N \rangle)\frac{\langle N \rangle^N}{N!}$, where N denotes the number of photons and $\langle N \rangle$ is the expectation of N.[47] The process of adding Poisson noise to PAI data depends on the input image's data type. N

corresponds to the photon count at each pixel, which is proportional to the pixel intensity in the image. For double-precision images, each pixel value is treated as the mean of a Poisson distribution after being scaled up by a factor of 1×10^{12} to convert the pixel values into a photon count domain prior to simulating Poisson noise, reflecting not an actual measurement of N, but rather a scaling step to facilitate modeling the noise profile. For instance, if a pixel has an initial value of 3.5×10^{-12}, it is first multiplied by 1×10^{12} to obtain 3.5. A Poisson-distributed value is then generated using this mean, and the result is subsequently scaled back down by 1×10^{12} to maintain consistency with the original intensity scale.

- *S&P*: This is another additive noise with a density of 5% for pixel destruction.[48]
- *Speckle*: This is a type of multiplicative noise with a uniform distribution having zero mean and 0.05 variance.[44,49,50]

These noise types with the above statistical values were chosen to simulate noise levels commonly encountered in experimental PAI scenarios while ensuring a sufficient SNR for meaningful analysis. In particular, these different types of noise variance correspond to a realistic level of background electronic and thermal noise typically observed in imaging systems. A higher variance would result in excessive degradation, making image reconstruction extremely challenging, while a lower variance would not sufficiently stress the denoising capability of the model.

C. Traditional non-learning noise removal algorithms

We explored five different types of non-learning noise removal algorithms in our study for comparative analysis with our DenP2P cGAN model concerning two image quality metrics—PSNR and SSIM.

- *Savitzky–Golay (SG) filtering:* This smoothening filter uses an SG finite impulse response filter with a defined polynomial order and a frame length.[51] We used MATLAB's inbuilt third order *sgolayfilt* function with a frame length of 11.
- *Wiener filtering:* This is a pixel-wise adaptive low pass filter[52] that works under the estimation of the local mean and variance calculated from a user-provided neighborhood of a certain dimension. We used the *wiener2* function in MATLAB with a 5×5 neighborhood window.
- *Total variation (TV) denoising:* This non-linear edge-preserving filter is based on the principle that the noise component of any image is generally highly spurious and thus has high TV, and the objective is to minimize the TV along with the sum of squared errors. We used a *TV-L1* image denoising model regularized with the primal–dual

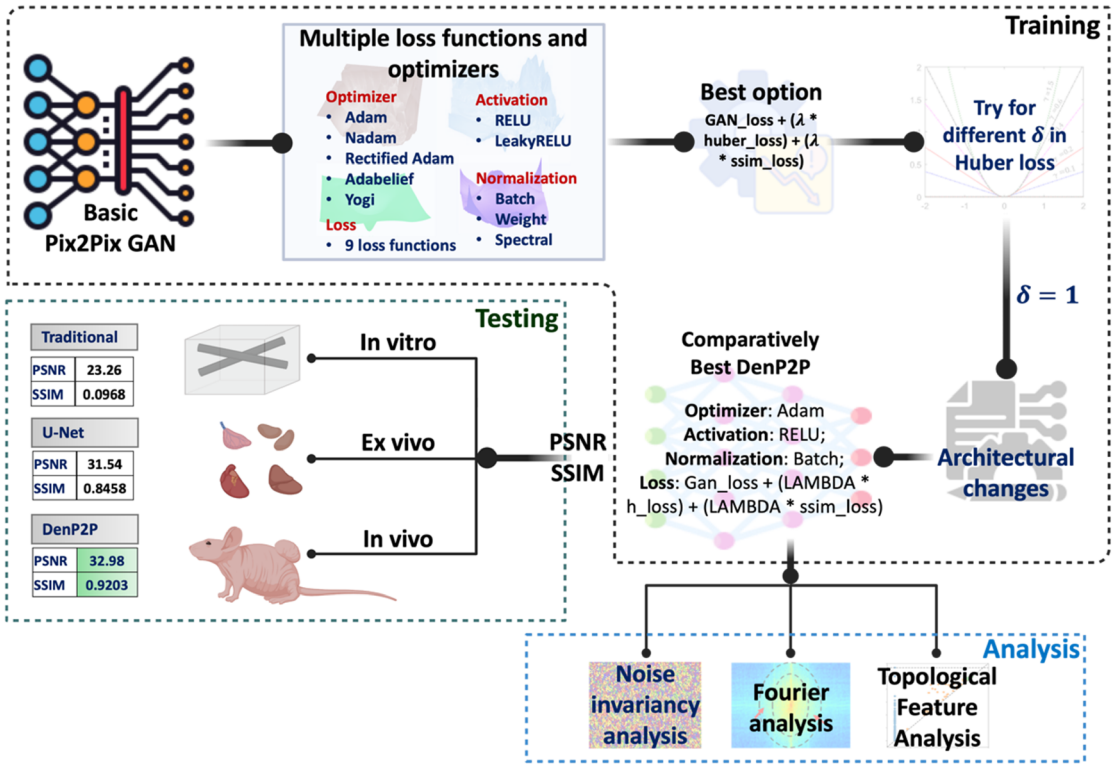

FIG. 1. Our approach toward developing an optimized DenP2P—whose source code base is taken from the conditional GAN Pix2Pix architecture.

algorithm.[53] The regularization parameter was set to 1, and it was iterated 400 times.

- *Non-local means (NLM):* It differs from local neighborhood comparison-based filtering with respect to the non-local averaging of all the pixels in an image.[54] First, the noise standard deviation from the noisy image (S_{est}) is estimated. Then, the parameters are chosen as $h = 1.15^{*} S_{est}$, fast_mode = True, *patch_size* = 5, and *patch_distance* = 3.
- *Block-matching 3D (BM3D):* This filtering is an advanced version of the NLM.[55] Like NLM, we estimated S_{est} and pass the S_{est} into the BM3D algorithm. We used *ALL_STAGES* where both the cascades—hard thresholding and Wiener filter stage—are performed, making it slow.

D. Deep learning networks

U-Net architecture

We considered the U-Net architecture[56] and modified the fourth block of the encoder and the bottleneck layer by adding a dropout (probability of 0.5) layer. An *Adam* solver (initial learning rate: 1×10^{-4}) with *MSE* loss was considered for the network. The encoder (contracting path) captures contextual information and progressively reduces spatial dimensions through layers of convolution, ReLU activation, and max-pooling. The decoder (expanding path) then restores spatial resolution using up-convolutions (transposed convolutions). At each up-sampling step, skip connections from corresponding encoder layers are merged with the decoder to retain fine-grained information from earlier stages. These skip connections allow U-Net to preserve spatial details and enhance localization, making it highly effective for applications. In one of our previous studies,[38] we found that the basic four-layered U-Net effectively addressed the enhancement of SNR as evidenced by improvements in PSNR and SSIM, which are not statistically significantly worse than the outcomes from the advanced U-Net algorithms. Therefore, for this investigation, we opted for the U-Net architecture.

DenP2P architecture

Inspired by the Pix2Pix image-to-image translator cGAN, whose detailed architectural information is available in Ref. 27, we modified some of its hyperparameters and architectures to fit our image-denoising problem in this study (see Fig. 1 for a pictorial representation). The modifications are as follows:

Hyperparameter modifications in generator:

Loss functions:

cGAN adversarial loss: cGAN aims to learn a mapping function $G_{DenP2P} : \{x, z\} \rightarrow y$ between the observed noisy input, x, and random noise vector, z, and the ground truth, y. The adversarial cost function of the cGAN is expressed as

$$\mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) = \mathbb{E}_{x,y}[\log D_{DenP2P}(x, y)] + \mathbb{E}_{x,z}[\log(1 - D_{DenP2P}(x, G_{DenP2P}(x, z)))],$$

where the objective of the network is to achieve minimal generator (G_{DenP2P}) adversarial loss and maximal discriminator (D_{DenP2P}) loss in a two-player game.

Thus, $G^{*}_{DenP2P} = \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P})$.

L1 loss:[57] It is the absolute pixel value difference between the predicted and ground truth images. Its mathematical form is defined as

$$\mathcal{L}_{L1}(G_{DenP2P}) = \frac{1}{N}\sum_{i=1}^{N} \left| y^{i}_{ground\_truth} - y^{i}_{predicted} \right|.$$

L2 loss:[57] L2 loss is the pixel value squared difference between the ground truths and predictions. Its mathematical formula is depicted as

$$\mathcal{L}_{L2}(G_{DenP2P}) = \frac{1}{N}\sum_{i=1}^{N} \left(y^{i}_{ground\_truth} - y^{i}_{predicted} \right)^{2}.$$

Sometimes, L1 loss is preferred over L2 loss due to its over-sensitivity (imposing larger penalty) for the outliers.

Huber loss:[58] This is a combination of L1 and L2 loss with a delta (δ) regulation value to make best out of both worlds. The equation is as follows:

$$\mathcal{L}_{H}(G_{DenP2P}) = \begin{cases} \frac{1}{2}\left(y^{i}_{ground\_truth} - y^{i}_{predicted} \right)^{2} & for \left| y^{i}_{ground\_truth} - y^{i}_{predicted} \right| \le \delta, \\ \delta \cdot \left[\left\| y^{i}_{ground\_truth} - y^{i}_{predicted} \right\|_{1} - \frac{1}{2}\delta \right], & otherwise. \end{cases}$$

SSIM loss:[59] This loss is concerned about the morphological similarity between the predictions and the ground truths. The SSIM loss is defined as

$$\mathcal{L}_{SSIM}(G_{DenP2P}) = \frac{1}{N}\sum 1 - SSIM\left(y^{i}_{ground\_truth} - y^{i}_{predicted} \right),$$

where SSIM is defined in Sec. II G.

PSNR loss: We also considered PSNR loss in our network architecture options, which keeps track of the maximum possible power of a signal difference between the predicted and labeled images. The loss is defined as

$$\mathcal{L}_{PSNR}(G_{DenP2P}) = \frac{1}{N}\sum \frac{1}{PSNR\left(y^{i}_{ground\_truth} - y^{i}_{predicted} \right)},$$

where PSNR is defined in Sec. II G.

Different combinations of loss functions: We tried nine different combinations of loss functions considering some or all the above-mentioned losses. As each loss type contributes to different aspects of image quality, combining them helps balance between pixel-wise accuracy, structural similarity, and perceptual realism. If the GAN's generator relies solely on the adversarial loss, it may lead to unrealistic artifacts or distortions because the discriminator only provides feedback about the realism of the image, not how numerically similar it is to the ground truth. By adding L1/L2/Huber, SSIM, or PSNR to the generator loss, we provide additional constraints that regularize the training, guiding the generator to produce images that are both perceptually realistic and numerically accurate to the ground truth.[60,61] In this case, our goal is to achieve a comprehensive loss that balances pixel-wise accuracy, structural preservation, and perceptual realism. Our network will ensure whether the generated image is numerically close to the ground truth through *L1, L2, or Huber* loss and via *SSIM* or *PSNR* loss, we can make sure that the outcomes have perceptual quality, with preserved structural integrity that is critical for maintaining image fidelity under noise or corruption. The objective functions with a combination of different loss functions are summarized as follows:

$$G^{*}_{DenP2P} = \textbf{Loss1} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L1}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss2} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L2}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss3} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{H}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss4} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L1}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss5} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L2}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss6} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{H}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss7} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L1}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{PSNR}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss8} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{L2}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{PSNR}(G_{DenP2P}), \text{ or}$$
$$\textbf{Loss9} : \arg min_{G_{DenP2P}} max_{D_{DenP2P}} \mathcal{L}_{cGAN}(G_{DenP2P}, D_{DenP2P}) + \lambda_{DL}\mathcal{L}_{H}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{SSIM}(G_{DenP2P}) + \lambda_{DL}\mathcal{L}_{PSNR}(G_{DenP2P}).$$

Final objective function: We experimented with three δ values for the Huber loss and fixed δ as 1. The final objective function, we considered in this work, is

$$\boldsymbol{G^{*}_{DenP2P}} = \textbf{arg}\boldsymbol{min_{G_{DenP2P}} max_{D_{DenP2P}}} \mathcal{L}_{cGAN}(\boldsymbol{G_{DenP2P}}, \boldsymbol{D_{DenP2P}}) + \boldsymbol{\lambda_{DL}}\mathcal{L}_{H}(\boldsymbol{G_{DenP2P}}) + \boldsymbol{\lambda_{DL}}\mathcal{L}_{SSIM}(\boldsymbol{G_{DenP2P}}).$$

We fixed the value of λ_{DL} to 100 after many hits and trails.

Optimizers:

- *Adam*: This is an extension to the classical stochastic gradient descent (SGD),[62] which considers the combination of advantages from AdaGrad[63] and RMSProp.[64] The initial learning rate, which was fixed for all network weights and non-changing over the whole training period for the case of SGD, is maintained for each network weight and adaptive over the training process in *Adam* optimization.[65] The initial learning rate (LR) was set to 1×10^{-4}, β_1 to 0.5, β_2 to 0.999, and ϵ to 10^{-7}.
- *Nadam*: This optimizer combines *Adam* and Nesterov momentum[66] for the learning update rule.[67] The LR was set to 1×10^{-4}, β_1 to 0.5, β_2 to 0.999, and ϵ to 10^{-7}.
- *Rectified Adam*: The rectified *Adam* aims to achieve *Adam*-like convergence with a smaller number of epochs, and its adaptive learning rate is rectified.[68] The LR was set to 1×10^{-3} and the minimum LR was set to 1×10^{-5} with a *warm_up* proportion of 0.1, and the total steps were taken as 40 000. Therefore, for the first 4000 steps, the LR was increased linearly from 0 to 1×10^{-3}, and then, it linearly reduced to reach the minimum LR at the end.
- *AdaBelief*: Another variant of *Adam* whose convergence is fast and SGD-like generalization. It is named so due to its adaptivity to the step size based upon its belief in the direction of the gradients.[69] We considered LR to be 1×10^{-4}, β_1 as 0.5, β_2 as 0.999, and ϵ as 10^{-14}.
- *Yogi*: Instead of using multiplicative exponential moving average-based adaptive methods in *Adam*, Yogi utilizes additive updates, which are beneficial for sparse nonconvex optimization problem space.[70] The LR was set to 1×10^{-4}, β_1 to 0.5, β_2 to 0.999, and ϵ to 0.001.

Architectural changes:

Activation layers:

We considered two types of activation layers—ReLU and leaky ReLU. They are most frequently used activation functions because they offer specific benefits that are crucial for the performance and stability of GANs. Due to ReLU's sparse activations, training our model is computationally efficient. We also do not encounter vanishing gradient problem. Leaky ReLU solves the zero gradient for negative inputs to avoid dead neurons, ensuring that all neurons continue to learn throughout training by achieving better gradient flow. The generator in a GAN often learns high-level abstractions, such as shapes, textures, and fine details. ReLU helps create these abstractions by keeping activations sparse, while leaky ReLU helps refine these features by ensuring that the network does not suffer from dead neurons, which would hinder learning specific image characteristics.[71,72] The careful balancing act between the generator and discriminator requires smooth and robust learning dynamics, which these activations help provide. We experimented with both *Adam* and *rectified Adam* solvers.

Normalization layers:

We tried three normalization layers in combination with the two activation layers.

- *Batch:*[73] Its main goal is to make the training process fast and stable by reducing internal covariate shifts and normalizing statistics of the pre-activation for each mini-batch of the training. It might also lessen the requirement for dropout layers.
- *Weight:*[74] This is a simple reparameterization of the weights to converge quickly, and it does not impose any dependence between the training examples of a mini-batch.
- *Spectral:*[75] This is another network weight normalization technique stabilizing the GAN training, where only one hyperparameter—Lipschitz constant—is tuned.

Final architecture of the generator:

We chose ReLU activation with batch normalization in each of the blocks of the encoder–decoder pathway of the generator, *Adam* solver, and the custom loss function as a combination of original adversarial loss, Huber, and SSIM loss with a respective weight factor (1, 100, 100). The total developmental workflow of the downstream optimized DenP2P model is displayed in Fig. 1. All our deep learning codes are implemented in TensorFlow Keras (version: 2.9.2) using a Tesla T4 GPU (CUDA version: 11.2, GPU VRAM: 25.45 GB).

E. Fourier spectrum analysis

We used the *fft2* command in MATLAB to translate the LA, HA, U-Net, and DenP2P generated images from the spatial domain to the frequency domain. For moving the zero-frequency component nearer the center of the Fourier converted frequencies, we employed *fftshift* to rearrange the frequencies. To prevent the zero-division issue, we changed the frequency array into the log scale for display purposes.

F. Topological feature analysis

A homology class represents an equivalence class of chains in a given space.[37] Mathematically, let X be a topological space. A k-dimensional chain in X is a formal sum of k-dimensional simplices with integer coefficients. A k-dimensional simplex is a geometric object that generalizes triangles, tetrahedra, and their higher-dimensional analogs. Now, consider the set $C_k(X)$ of all k-dimensional chains X. The boundary operator $\partial_k : C_k(X) \rightarrow C_{k-1}(X)$ maps a k-dimensional chain to its boundary. The kth homology of group X, denoted by $H_k(X)$, is defined as the quotient group: $H_k(X) = \frac{\ker(\partial_k)}{im(\partial_{k+1})}$, where $\ker(\partial_k)$ is kernel of the boundary operator ∂_k and $im(\partial_{k+1})$ is the image of the boundary operator ∂_{k+1}. The elements $H_k(X)$ are called k-dimensional homology classes. Each homology class represents a cycle, which is a closed k-dimensional object, modulo the boundary of a higher-dimensional object. Now, the key idea behind persistent homology is to construct a filtration of the space by successively adding and removing simplices (geometric objects of various dimensions) based on distance. At each step of the filtration, the homology groups of the space are computed, resulting in a sequence of homology groups. By tracking how the homology groups change over the course of the filtration, persistent homology identifies topological features such as connected components, loops, voids, and higher-dimensional voids. The persistence of these features is measured by their lifespan, which represents the range of parameter values over which the feature exists.

A persistence diagram is a visual representation of the topological features present in the data, typically computed using the *Vietoris–Rips* complex[76,77] in persistent homology. Each point in the persistence diagram represents its birth (x-coordinate) and death (y-coordinate) values. These values represent the lifespan of the topological feature. The diagonal line represents the points where a feature is born and dies at the same time, indicating noise or spurious features in the data. The points near the diagonal line represent short-lived features or noise, and the points far from the diagonal line represent significant topological features that persist across different scales. Features with higher persistence values are more stable and robust. To calculate the homology of an image in *Python*, we used a lean persistent homology package—*ripser* (*scikit-tda*)—whose core computational engine is run on the C++ *Ripser* package.[78] We considered maximum homology dimension as 2 (H_0, H_1, and H_2), entire filtration, coefficients in the prime field Z/2Z, cosine metric to compute the distance, and no number of points to subsample in "greedy permutation." For plotting, we used the *persim.plot_diagrams* helper function to plot persistence diagrams.

G. Image quality metrics

To check the quality of the cGAN-generated output images, we used two full reference image quality (IQA) metrics, which are generally believed to be critical parameters in the assessment.

PSNR: It is a full reference quality metric[79] measured in dB scale, which is a ratio signal and mean squared error (MSE) into account and is expressed in logarithmic terms because signals sometimes might have a dynamic wide range. The PSNR is defined as

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

where MAX_I is the maximum possible pixel value of image I.

SSIM: Structural similarity index (SSIM)[80] is another full reference image quality metric ranging between 0 and 1, which measures the amount of distortion in a reconstructed image compared to the ground truth. The SSIM is defined as

$$\boldsymbol{SSIM}(\boldsymbol{A},\boldsymbol{B}) = \frac{\left(2\boldsymbol{\mu_A}\boldsymbol{\mu_B} + \boldsymbol{c}_1\right)\left(2\boldsymbol{\sigma_{AB}} + \boldsymbol{c}_2\right)}{\left(\boldsymbol{\mu_A^2} + \boldsymbol{\mu_B^2} + \boldsymbol{c}_1\right)\left(\boldsymbol{\sigma_A^2} + \boldsymbol{\sigma_B^2} + \boldsymbol{c}_2\right)},$$

where $\boldsymbol{\mu_A}$ is the sample mean of $\boldsymbol{A}$, $\boldsymbol{\sigma_A^2}$ is the variance of $\boldsymbol{A}$, $\boldsymbol{\sigma_{AB}}$ is the covariance of $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\boldsymbol{c}_1$ ($[\boldsymbol{k}_1\boldsymbol{L}]^2$) and $\boldsymbol{c}_2$ ($[\boldsymbol{k}_2\boldsymbol{L}]^2$) are determined based on $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, which are set as 0.01 and 0.03, respectively, and $\boldsymbol{L}$ is the dynamic range.

TABLE I. Combinations of various optimization and loss functions for the DenP2P generator showing the performance with respect to PSNR (dB).

	Adam	Nadam	Rectified Adam	AdaBelief	Yogi
Loss1	30.5139	30.2758	31.8329	30.7737	30.0428
Loss2	29.8139	30.6823	31.1939	30.605	29.5477
Loss3	31.598	29.4842	30.3619	31.2937	28.945
Loss4	32.0915	32.2839	32.0072	31.9401	32.307
Loss5	32.4821	31.8475	32.1084	30.8784	31.7502
Loss6	32.9775	32.7205	32.8932	32.5826	32.2336
Loss7	31.8414	31.0985	31.9006	32.6096	31.6035
Loss8	25.13	31.4624	32.2692	27.4118	32.4259
Loss9	31.3239	31.6663	31.9433	32.375	31.4827

TABLE II. Combinations of various optimization and loss functions for the DenP2P generator showing the performance with respect to SSIM (a.u.).

	Adam	Nadam	Rectified Adam	AdaBelief	Yogi
Loss1	0.8543	0.8091	0.8853	0.8362	0.8025
Loss2	0.7056	0.7389	0.7542	0.7171	0.7917
Loss3	0.7406	0.7231	0.7491	0.7387	0.7403
Loss4	0.9043	0.9046	0.9074	0.9054	0.9176
Loss5	0.9109	0.9049	0.9069	0.9048	0.9033
Loss6	0.9203	0.9218	0.9168	0.9186	0.9083
Loss7	0.9075	0.8876	0.9085	0.9207	0.9009
Loss8	0.868	0.8885	0.9067	0.8746	0.9084
Loss9	0.9032	0.9025	0.897	0.9047	0.8979

III. RESULTS AND DISCUSSIONS

A. Effects of various hyperparameter modifications

We implemented the original Pix2Pix network architecture and then tried different optimizers and loss function hyperparameter combinations in the generator part as shown in Tables I and II, respectively. These tables showcase the values of PSNR and SSIM, respectively. The bold-faced value in each column denotes the best image quality metric measurement among all the loss functions for a specific optimizer. Both the PSNR and SSIM values are consistently high for loss 6, which is a combination of generator loss, Huber, and SSIM loss. We charted out box plots of PSNR and SSIM values in Fig. 2 for all the optimizers considering all the nine loss functions.

Among the loss function combinations tested, Adam and rectified Adam optimizers consistently performed satisfactorily across every test case. Moreover, they performed the best when we considered loss 6. Although there was no statistically significant difference between the optimizers (confirmed by the unpaired parametric two-tailed t-test with Welch's correction), Adam was preferred. Adam's stability and robustness have been validated in various settings, making it a safe choice for different kinds of problems. Some of the alternatives such as Nadam, rectified Adam, and Yogi add slight computational overhead (93 s per 1000 steps for Adam compared to 126, 109, 121, 115 s per 1000 steps for Nadam, rectified Adam, AdaBelief, and Yogi, respectively) due to additional components (e.g., momentum or rectification steps). Faster convergence implies that fewer iterations are required to achieve the same or better results. This is advantageous in resource-constrained environments where computational efficiency and time are critical. For applications requiring frequent retraining (e.g., real-time denoising or clinical setups), Adam's faster convergence could lead to more efficient use of computational resources. In addition, Adam's simpler parameter tuning makes it more user-friendly and easier to implement. Optimizers such as AdaBelief or Yogi introduce additional hyperparameters that may require careful tuning depending on the dataset and problem. Adam's simplicity reduces overhead in experimentation, especially when the outcomes (PSNR and SSIM) are already comparable. Overall, clinically translatable models need to generalize beyond just denoising, perhaps to related tasks such as segmentation or even classification (as part of multi-task learning), and Adam might offer reliable generalization for those cases.

TABLE III. Adjusting δ parameter for the Huber loss function of the DenP2P generator showing the performance with respect to PSNR and SSIM.

	PSNR	SSIM
$\delta = 0.5$	31.3633	0.9013
$\delta = 1$	32.9775	0.9203
$\delta = 1.5$	31.1087	0.8939

TABLE IV. Combinations of architectural modifications for the DenP2P generator showing the performance with respect to PSNR and SSIM.

	Adam		Rectified Adam	
Activation + Normalization	PSNR	SSIM	PSNR	SSIM
ReLU + batch	32.9775	0.9203	32.8932	0.9168
ReLU + weight	32.6144	0.9143	31.8501	0.8964
ReLU + spectral	32.2567	0.9024	30.7953	0.9076
Leaky ReLU + batch	32.1027	0.9098	32.3152	0.9114
Leaky ReLU + weight	31.7771	0.9078	32.046	0.9012
Leaky ReLU + spectral	32.405	0.9037	31.3528	0.8952

We also examined the Huber loss function, with different options for the δ value (0.5, 1, or 1.5), which acts as a control parameter between L1 and L2 loss functions. Table III represents the corresponding PSNR and SSIM values for the Adam solver with the combination loss 6 except the variation of value for the Huber loss. We observed that the optimal δ value is 1, as it yielded the highest values for both PSNR and SSIM among all the options. In addition, the performance of rectified Adam was noteworthy. Although it did not yield the highest PSNR and SSIM, it demonstrated stable performance across all the loss functions, with the smallest standard deviation.

B. Effects of various architectural modifications

After determining the optimal hyperparameter options, this study investigated the influence of architectural constructs, specifically activation and normalization layers, as outlined in Sec. II.

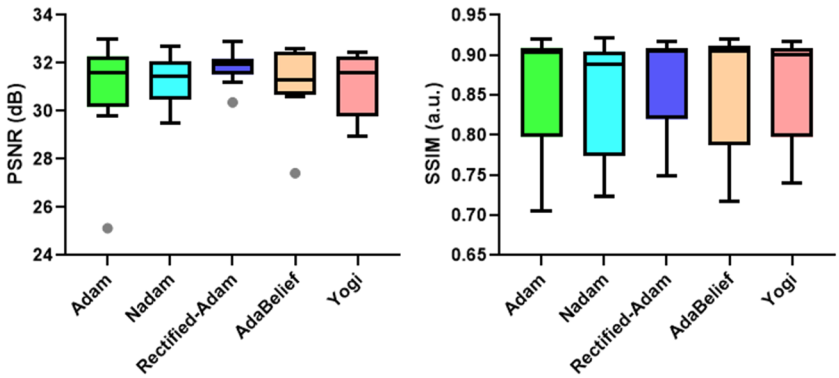

FIG. 2. Optimizer (*Adam*) selection with respect to PSNR and SSIM out of all the other advanced optimizers for all the losses. The image quality metrics are not significantly improved with other optimizers than with *Adam*.

Similar to the hyperparameter modifications, we also catered to modify some of the architectural constructs of the basic Pix2Pix by fitting different activation functions (ReLU[81] and leaky ReLU[82]) and normalization layers (batch, weight, and spectral). In total, we tried six combinations to measure PSNR and SSIM. Table IV depicts the outcomes for two equally best optimizers and the combination loss 6 in six different architectural combinations in each row.

Adam and rectified Adam showed similar performance across the IQA metrics (Table IV), leading to the selection of ReLU activation and batch normalization layers. Both Adam and rectified Adam are variants of adaptive learning rate optimizers, meaning that they adjust learning rates based on past gradients. They use first-order (mean) and second-order (variance) moment estimates, making them highly effective for a wide range of tasks. This shared foundation ensures that both Adam and rectified Adam optimize similarly in many cases. Rectified Adam introduces a "rectification term" that helps smooth out learning in the early stages of training, and our denoising problem does not suffer from these initial instability issues due to well-behaved data. The rectification might not offer a significant performance boost.

C. Comparison with traditional non-learning methods

After fixing the optimal network parameters and architectural setup, we compared the performance of our DenP2P deep model with the traditional non-learning algorithms for PSNR and SSIM. Figure 3 shows three mouse tumor images at low and high number of frames averaging in the first and second columns, respectively, whereas the rest of the column images depict one of the five non-learning algorithm outputs except the last column that denotes DenP2P outcomes.

The respective image quality metric values are shown in Table V organized in a similar column fashion as done in the figure. The network outperformed traditional non-learning algorithms, as demonstrated in Fig. 3 and Table V, with respect to both evaluation metrics. This observation is supported by the fact that deep learning-based models, particularly convolutional neural networks (CNNs) or U-Net variants, can automatically learn representations of the underlying structure of the input data. These models excel at capturing complex patterns and relationships within the images through training, which traditional algorithms struggle with. During training, these networks learn to generalize across different noise levels, image structures, and even varying datasets. The deep learning networks can model highly non-linear functions, enabling them to effectively learn complex mappings between noisy and clean images. Instead of being limited by pre-programmed heuristics (as is the case with traditional methods), deep learning networks dynamically adjust their parameters based on the training data, thus achieving better denoising across diverse conditions. Traditional methods require an *a priori* understanding of the noise characteristics (e.g., variance in Gaussian noise) to tune their parameters appropriately. While effective for the noise they are designed to handle, these algorithms typically perform suboptimal when the image contains complex or multiple types of noise, as their fixed filters and models lack the ability to adapt. The generalization ability allows deep learning models to perform well across a wide range of noise types and intensities, making them robust and versatile compared to traditional methods.

TABLE V. Comparison of PSNR and SSIM between non-learning traditional algorithms and DenP2P.

	SG	Wiener	TV	BM3D	NLM	Pix2Pix
PSNR	22.7893	22.9701	23.2604	23.1501	22.2932	32.9775
SSIM	0.0856	0.0883	0.0968	0.0891	0.0859	0.9203

D. Performance across biological tissue

1. Comparison with U-Net

Figure 4 displays three representative *in vivo* tumor images, where the first column shows a low number of frame-averaged images, the second column denotes the ground truth, the third column displays the U-Net outcomes, and the last column shows our DenP2P outputs. Low frame-averaged images are noisy and of lower quality due to the limited number of averaged frames. Frame averaging is commonly used to enhance signal quality in imaging

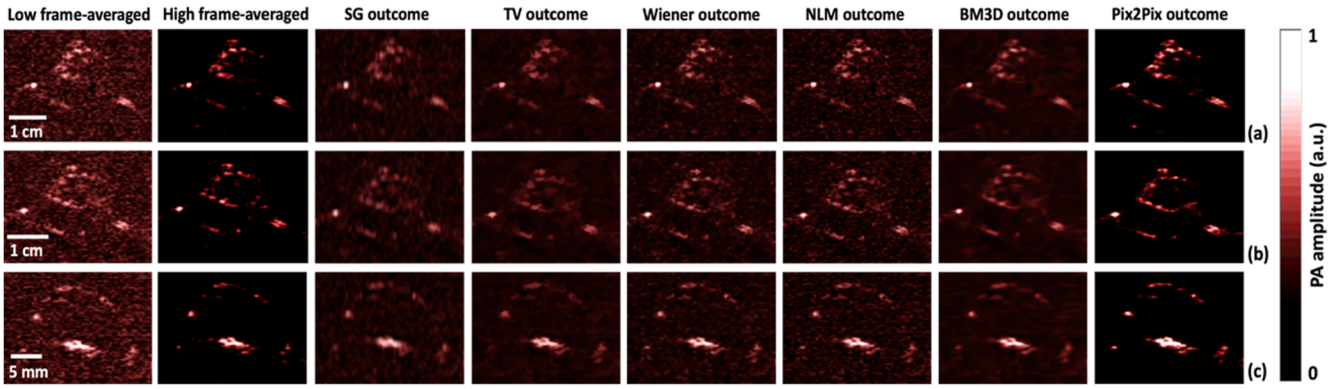

FIG. 3. Comparison between non-learning (SG, TV, Wiener, NLM, and BM3D) algorithms' outcomes and our deep learning DenP2P noise removal network's outcome. The rows (a)–(c) display data for three different cross sections of mouse tumors. All the scale bars are mentioned in the corresponding US images.

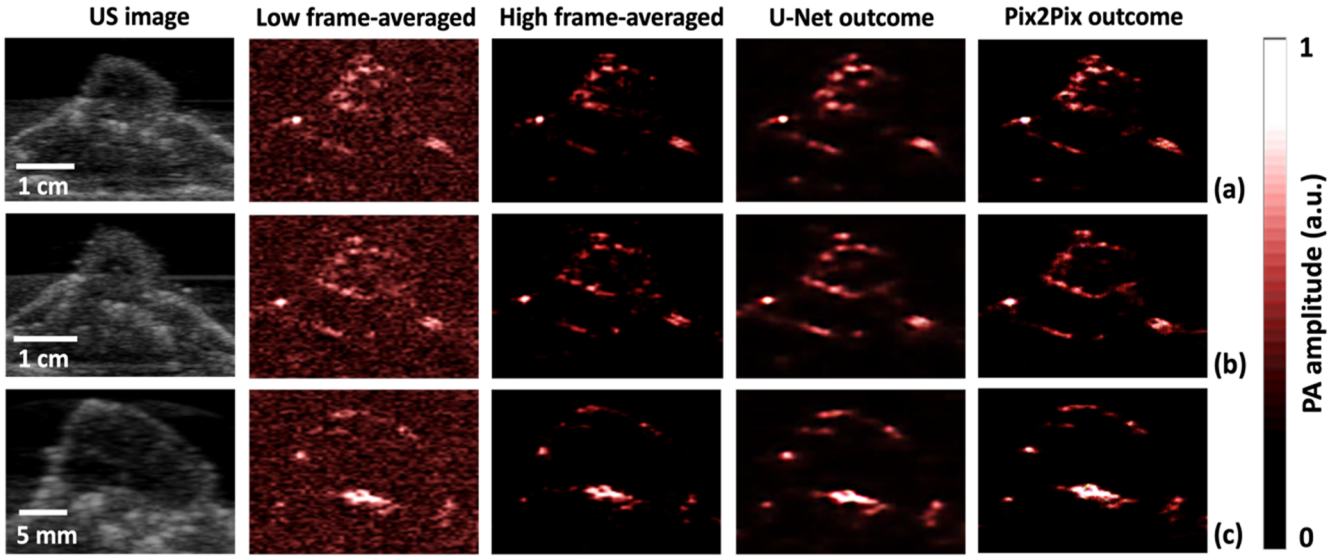

FIG. 4. Comparison between U-Net outcomes and our deep learning DenP2P noise removal network's outcome. The rows (a)–(c) display data for three different cross sections of mouse tumors. All the scale bars are mentioned in the corresponding US images.

by reducing random noise, but with fewer frames, the noise level remains high, reducing image quality. We considered high frame-averaged images as the ideal or reference ground truth images, which are either noise-free or represent the most accurate depiction of the target tissue. The U-Net column images serve as a benchmark to compare how well DenP2P performs. The DenP2P column outputs are expected to perform U-Net by better recovering image details, reducing noise, or enhancing features with greater accuracy.

TABLE VI. Comparison of PSNR and SSIM among several U-Net-based architectures and the DenP2P generative model.

	U-Net	U-Net++	Dense U-Net	Res U-Net	Attention U-Net	Attention Res U-Net	R2 U-Net	DenP2P
PSNR	31.5381	29.9587	31.8746	29.8256	30.1440	31.0613	31.8382	32.9775
SSIM	0.8458	0.8432	0.8638	0.8012	0.8132	0.8147	0.8574	0.9203

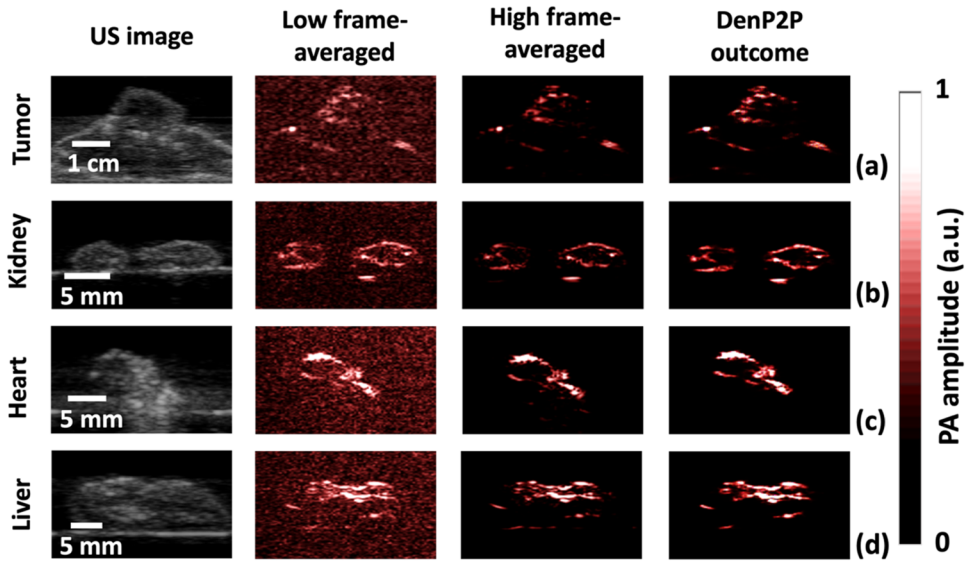

FIG. 5. DenP2P outputs for (a) *in vivo* mouse tumor and *ex vivo* organs such as (b) kidney, (c) heart, and (d) liver, in comparison with the ground truth HA images.

Table VI presents a quantitative analysis of the image outputs, comparing U-Net and DenP2P on specific image quality metrics. The PSNR measures the noise level in the reconstructed image compared to the reference ground truth, whereas the SSIM indicates preservation of the image's structural features. By comparing these metrics, Table VI objectively shows whether DenP2P provides significant improvements over U-Net, particularly in the context of image denoising and feature preservation. There are other complex CNN-based models with attention, residual, R2, and dense connection layers in the encoder–decoder architectures, which are heavily used in computer vision.[38] We are also providing their corresponding quality metric values for comprehensive benchmarking.

2. In vivo and Ex vivo performance of DenP2P

Figure 5 expands on the evaluation by providing a broader set of images, encompassing both *in vivo* (living tissue) and *ex vivo* (extracted organs) imaging. This setup enables a thorough assessment of the model's generalizability across different types of tissues and imaging conditions. The *in vivo* images allow the evaluation of DenP2P's ability to process live, dynamic tissue with inherent motion or physiological variability, which is typically more challenging for denoising algorithms due to variations. The *ex vivo* images of organs are often more static, making it easier to focus on noise reduction without the complication of motion artifacts. DenP2P's performance on these images helps demonstrate its robustness across different types of tissue and organ systems.

The SSIM metric did not yield satisfactory results for the U-Net, possibly due to blurring, which has been discussed in detail in Sec. III F. Our denoising model was tested with a variety of *ex vivo* biological organs and *in vivo* mouse tumors, as shown in Fig. 5. These test data samples were morphologically different from the training examples, indicating that our model can handle out-of-class test data distributions that differ spatially from the training sets.

TABLE VII. Combinations of architectural modifications for the DenP2P generator showing the performance with respect to PSNR and SSIM.

	Adam		Rectified Adam	
Noise distribution types	PSNR	SSIM	PSNR	SSIM
Gauss	30.666	0.8814	30.4414	0.894
Poisson	34.0497	0.929	34.0069	0.9355
Salt and pepper	30.18	0.831	29.682	0.8061
Speckle	32.4801	0.9158	33.2879	0.9287

E. Noise distribution type invariancy

To evaluate the noise distribution type invariance of our deep learning network, we conducted experiments using DenP2P on images corrupted with different artificially generated noise types, namely Gaussian, Poisson, S&P, and speckle. By testing DenP2P on these noise types, the goal is to demonstrate how well the model generalizes across varying noise distributions, showing that it can adapt to different noise patterns without specialized tuning. Figure 6 illustrates the various noise-corrupted scenarios of a mouse tumor, where each row represents a different noise type. The columnar arrangement of this figure remains consistent with that of Fig. 5 except the first column having US image. This visual representation allows for a direct comparison of how different noise types affect the tumor image and how effectively DenP2P denoises across these different noise-type scenarios.

We also calculated PSNR and SSIM values for both *Adam* and *rectified Adam* solvers, which are given in Table VII.

In our previous study,[25] we observed that the denoising U-Net struggled with S&P noise, despite effectively removing Gaussian noise. However, our DenP2P model demonstrates strong denoising performance across Gaussian, Poisson, and speckle noise, indicating its ability to generalize across these noise distributions, as depicted in

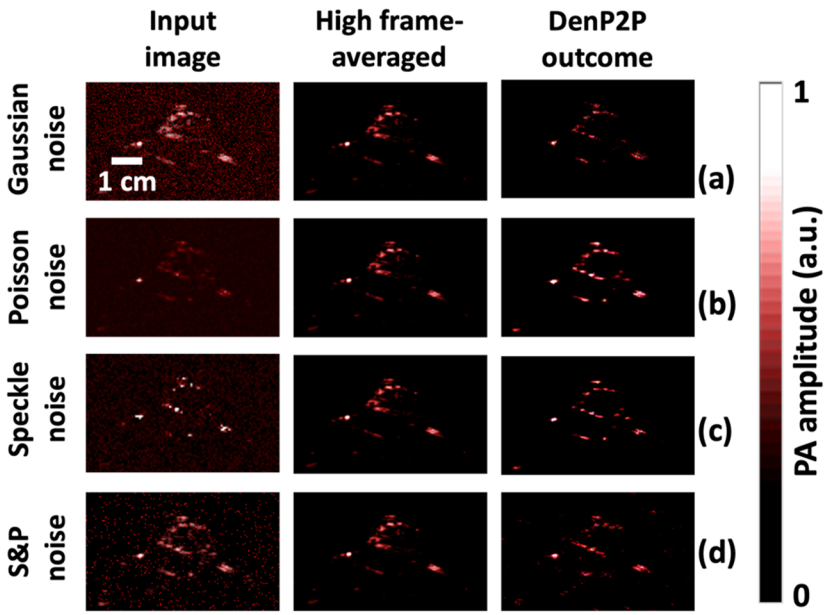

FIG. 6. Noise invariancy test—we corrupt the high number of frame images shown in the second column with (a) Gaussian, (b) Poisson, (c) speckle, and (d) S&P noise, respectively, depicted in the first column. The corresponding DenP2P outcomes are provided in the last column.

Fig. 6 and Table VII. Salt and pepper noise is particularly challenging because it corrupts random pixels with extreme values, making it harder for a neural network to detect and remove. Even though DenP2P shows a degree of noise-invariance, S&P noise leads to a slight drop in SSIM due to the nature of the noise corrupting fine details and edges, which are crucial for structural similarity. Nevertheless, DenP2P still outperforms U-Net in handling S&P noise, demonstrating better adaptability. The choice of optimizer (whether Adam or rectified Adam) did not significantly affect the model's noise-invariance. This suggests that the architecture of DenP2P and the learning dynamics imposed by the adversarial framework play a more crucial role in achieving noise-invariance than the specific optimizer used for gradient updates. While U-Net is highly effective at removing structured and predictable noise (such as Gaussian noise, which is normally distributed and easy to filter with learned features), it struggles with S&P noise due to its impulsive nature, where random pixels are corrupted. However, for the DenP2P model, the adversarial loss, combined with the pixel-wise loss functions, helps the network generalize across varying noise types. The adversarial process forces the generator to capture the true underlying structure of images, independent of the noise characteristics. Unlike U-Net, which may overfit to specific noise types seen during training, DenP2P implicitly models the noise distribution by focusing on reconstructing the clean image itself rather than explicitly modeling the noise. This allows it to adapt to various noise types without being constrained by assumptions about the noise's distribution.

F. Frequency domain analysis

In Fig. 7, we present the frequency magnitude spectrum analysis of the results obtained from U-Net and DenP2P in the Fourier domain for two representative *in vivo* mouse tumors. The first and third rows depict the spatial domain images of the mouse tumors, while the second and last rows display the corresponding Fourier domain plots. The first two columns represent the images obtained using LA and HA, respectively, and the last two columns show the outcomes of the U-Net and DenP2P models, respectively. By

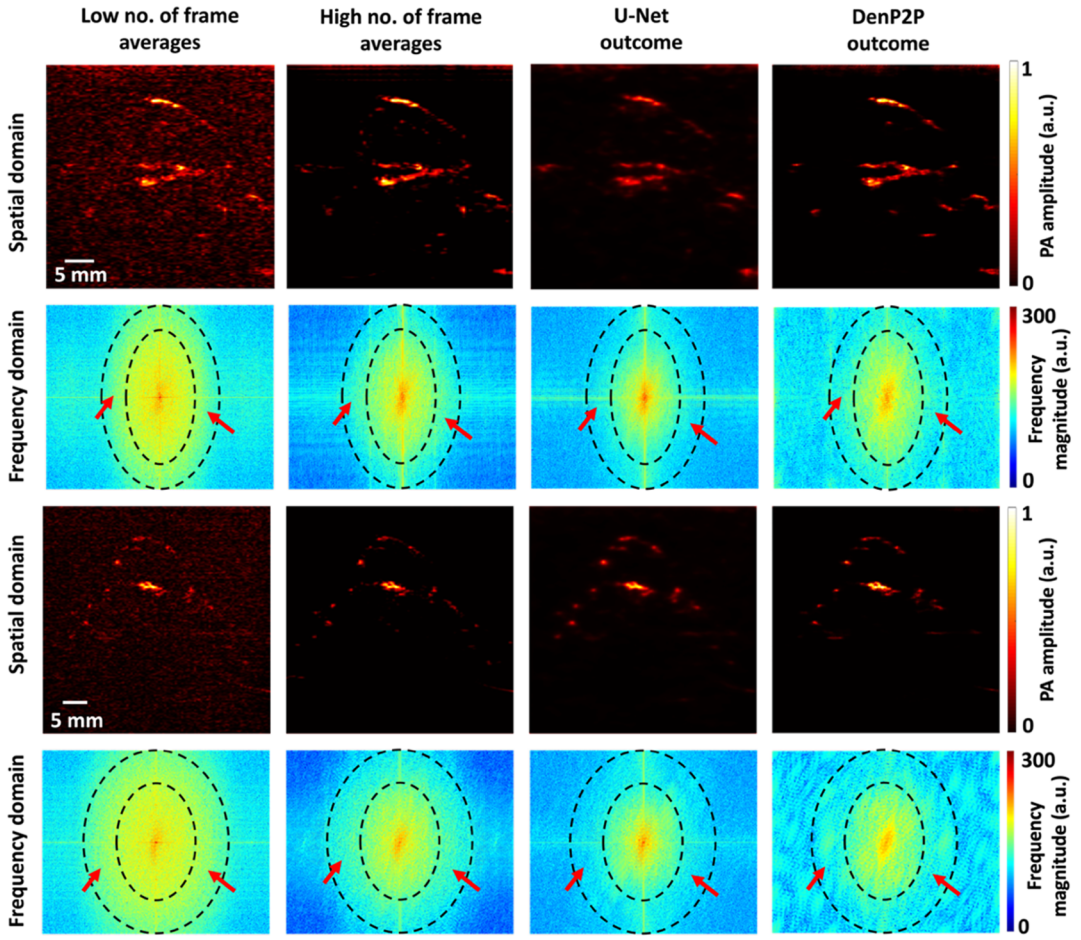

FIG. 7. Frequency magnitude spectrum analysis—comparison among low (first column) and high (second column) number of frame averaged images, U-Net outcomes (third column), and the DenP2P outputs (last column) for tumor images of two mice.

comparing U-Net and DenP2P in the frequency domain, we found that the high-frequency magnitude spectrum was not reduced for DenP2P outcomes, unlike U-Net generated images. Even the high number of frame averaging images exhibited a loss of high diagonal frequency components, as confirmed in Fig. 7. To quantitatively assess the blurring of the deep network outputs compared to the ground truths (high number of frame averaging), we employed a method mentioned in Ref. 83. The analysis revealed that U-Net decreased sharpness by 25.5 ± 4.99%, whereas our DenP2P network's outcomes were nearly blur-free, with only a 1.25 ± 0.98% reduction in sharpness. In addition, the cGAN outcomes did not exhibit noticeable edge effects, suggesting that it is highly effective in maintaining the integrity of edges and transitions in the image. This is particularly important in medical imaging, where clear boundaries between anatomical structures are critical for accurate diagnosis.

U-Net models typically rely on pixel-wise loss functions. These losses emphasize global similarity but can lead to over-smoothing as the model prioritizes minimizing overall pixel error over preserving high-frequency details. DenP2P, on the other hand, uses an adversarial loss that encourages the generator to produce images indistinguishable from real ones. This process leads to better preservation of high-frequency details as the discriminator penalizes blurring or loss of fine structures, helping DenP2P maintain sharpness. The adversarial training also forces the generator to match the high-frequency distribution of the ground truth images, preventing over-smoothing.

G. Topological analysis

Figure 8 depicts a concise and informative summary of the topological features present in LA (first column), HA (second column), and the DenP2P outcomes (last column) for a representative subcutaneous tumor cross section for which we calculated the persistent homology classes up to H2. For the second row, each plot's horizontal axis represents the initiation or birth of a particular feature, and the vertical axis denotes the disappearance or death of that feature. The last row shows the corresponding lifetime of the topological features.

The persistence homology diagrams in Fig. 8 substantiate the fact that LA images consist of extraneous noisy topological characteristics within LA images, characterized by short lifespans, resulting in a concentration of H_1 class points near the diagonal lines. Conversely, both HA and DenP2P outputs demonstrate a notable reduction in noisy topological features, evidenced by the sparsity of extraneous features along the diagonal lines. The lifetime plots in Fig. 8(c) also depict the same quantitative conclusions. The highly

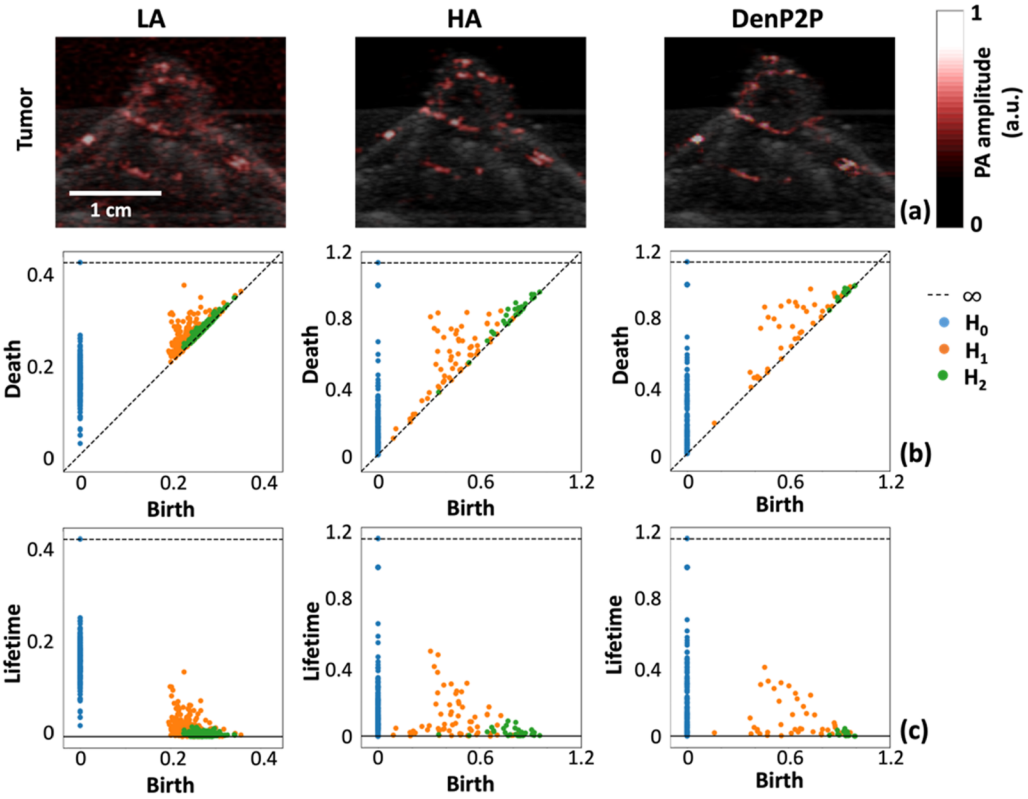

FIG. 8. Persistence diagrams up to H_1 homology dimension for low (LA) and high (HA) number of frame-averaging and DenP2P outcomes. (a) Representative tumor image for which the persistence homologies are calculated. (b) Persistence diagrams for HA, LA, and DenP2P, respectively, up to homology class H_3. (c) Corresponding lifetime plots for the topological features. N.B. All the images are ultrasound and PA co-registered.

prominent or salient topological structures in the data are mainly emerged from the deep learning outcomes. Contrasted with LA persistence, the HA and DenP2P plots reveal delayed emergence (late birth) of certain H_1 class features, suggestive of potentially more intricate or expansive topological structures, such as significant loops, compared to features with earlier birth. Generally, features with late births tend to have higher persistence and greater stability compared to features with early births.[35] They persist across a broader range of filtration thresholds, indicating that they are more robust and less sensitive to minor variations in the data as they capture essential structural elements of the data underlying geometry or topology.

IV. CONCLUSION

Advancements in optical illumination, particularly LASER and LED technology, have greatly contributed to the development of PAI. This hybrid biomedical imaging modality combines the advantages of optical contrast and ultrasound depth resolution, making it a promising tool for various applications. Furthermore, the affordability and portability of LED-based PAI systems have made them suitable for low resource settings. However, one of the challenges faced in LED-based PAI is the low SNR of the captured images, primarily caused by the limited energy of the illumination. While we can achieve a high SNR through extensive frame averaging, this approach is not preferable for real-time imaging, which is crucial for analyzing the dynamics of vascular diseases and avoiding artifacts resulting from the body's natural movements, such as breathing and heartbeats. To address this challenge, we developed an optimized deep learning model called cGAN DenP2P, based on the Pix2Pix framework. Through extensive experimentation involving modifications of both hyperparameters and network architecture, we refined the DenP2P model to achieve superior performance compared to traditional non-learning methods and U-Net denoising algorithms. We used two comprehensive image quality metrics—PSNR and SSIM. We tested our optimized DenP2P model with *ex vivo* and *in vivo* biological organs that exhibited spatial variations from the training data. Remarkably, the model demonstrated noise distribution type invariance, indicating its ability to handle different types of noise. In addition, a frequency domain analysis revealed that our model's outcomes remained free from distortion, blur, and edge effects. Moreover, this study employed persistence homology diagrams to analyze the topological characteristics of the image data. The results highlighted the effectiveness of DenP2P in reducing noisy topological features, leading to the emergence of more stable and robust topological structures. One of our future objectives is to develop a system capable of learning from unpaired databases, as obtaining corresponding image-label pairs can be challenging in real-life scenarios. We will explore the potential of optimizing the rectified Adam optimizer, which demonstrates the least standard deviation despite not achieving the highest image quality metric values. We will also implement deep learning frameworks that can leverage complementary information from multiple imaging modalities to enhance denoising performance involving cross-modal consistency training that jointly optimizes denoising across different imaging techniques. We also observed the occasional artificial enhancement of intensities and the introduction of artifacts, which can impact the reliability of the reconstructed images as observed in Figs. 4 and 5. In addition, the model struggles to enhance or restore low-intensity signals effectively in some cases. Future work should focus on increasing the dataset size, incorporating more phantom variations, especially from different experimental conditions (e.g., more gain settings, depths, tissue types) to improve generalizability, and exploring advanced architectures, such as physics-informed neural networks or uncertainty-aware models, to mitigate the issues. Furthermore, post-processing techniques and hybrid approaches integrating traditional signal processing with deep learning may enhance the model's performance and ensure more accurate and reliable reconstructions.

ACKNOWLEDGMENTS

The authors acknowledge the funding support from Tufts School of Engineering and Tufts Data Intensive Science Center Pilot grant. The authors thank Mr. Marvin Xavierselvan for help with tumor implantation and Ms. Allison Sweeney for handling animal care.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Avijit Paul: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (equal); Writing – original draft (lead); Writing – review & editing (equal). **Srivalleesha Mallidi**: Conceptualization (supporting); Data curation (supporting); Formal analysis (supporting); Funding acquisition (lead); Investigation (supporting); Methodology (supporting); Project administration (lead); Resources (lead); Software (supporting); Supervision (lead); Validation (supporting); Visualization (supporting); Writing – original draft (supporting); Writing – review & editing (supporting).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] A. G. Bell, "Upon the production and reproduction of sound by light," J. Soc. Telegraph Eng. **9**(34), 404–426 (1880).
[2] M. Xu and L. V. Wang, "Photoacoustic imaging in biomedicine," Rev. Sci. Instrum. **77**(4), 041101 (2006).
[3] P. Beard, "Biomedical photoacoustic imaging," Interface Focus **1**(4), 602–631 (2011).
[4] L. V. Wang and S. Hu, "Photoacoustic tomography: In vivo imaging from organelles to organs," Science **335**, 1458–1462 (2012).
[5] Y. Zhu *et al.*, "Towards clinical translation of LED-based photoacoustic imaging: A review," Sensors **20**(9), 2484 (2020).
[6] M. Xavierselvan, M. K. A. Singh, and S. Mallidi, "In vivo tumor vascular imaging with light emitting diode-based photoacoustic imaging system," Sensors **20**(16), 4503 (2020).
[7] R. Bulsink *et al.*, "Oxygen saturation imaging using LED-based photoacoustic system," Sensors **21**(1), 283 (2021).

[8]M. Xavierselvan and S. Mallidi, *Led-Based Functional Photoacoustics—Portable and Affordable Solution for Preclinical Cancer Imaging* (Springer, Singapore, 2020).
[9]G. Paltauf, R. Nuster, and M. Frenz, "Progress in biomedical photoacoustic imaging instrumentation toward clinical application," J. Appl. Phys. **128**(18), 180907 (2020).
[10]M. Mafi *et al.*, "A comprehensive survey on impulse and Gaussian denoising filters for digital images," Signal Process. **157**, 236–260 (2019).
[11]A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists* (O'Reilly Media, Inc., 2018).
[12]L. Peng *et al.*, "The advances and challenges of deep learning application in biological big data processing," Curr. Bioinf. **13**(4), 352–359 (2018).
[13]M. Pandey *et al.*, "The transformational role of GPU computing and deep learning in drug discovery," Nat. Mach. Intell. **4**(3), 211–221 (2022).
[14]H. Deng *et al.*, "Deep learning in photoacoustic imaging: A review," J. Biomed. Opt. **26**(04), 040901 (2021).
[15]J. Gröhl *et al.*, "Deep learning for biomedical photoacoustic imaging: A review," Photoacoustics **22**, 100241 (2021).
[16]C. Yang *et al.*, "Review of deep learning for photoacoustic imaging," Photoacoustics **21**, 100215 (2021).
[17]H. Zhao *et al.*, "Deep learning enables superior photoacoustic imaging at ultralow laser dosages," Adv. Sci. **8**(3), 2003097 (2021).
[18]J. Kim *et al.*, "Deep learning acceleration of multiscale superresolution localization photoacoustic imaging," Light: Sci. Appl. **11**(1), 131 (2022).
[19]N. Awasthi *et al.*, "Deep neural network-based sinogram super-resolution and bandwidth enhancement for limited-data photoacoustic tomography," IEEE Trans. Ultrason., Ferroelectr., Freq. Control **67**(12), 2660–2673 (2020).
[20]P. Rajendran and M. Pramanik, "Deep learning approach to improve tangential resolution in photoacoustic tomography," Biomed. Opt. Express **11**(12), 7311–7323 (2020).
[21]N. Davoudi, X. L. Deán-Ben, and D. Razansky, "Deep learning optoacoustic tomography with sparse data," Nat. Mach. Intell. **1**(10), 453–460 (2019).
[22]A. Hariri *et al.*, "Deep learning improves contrast in low-fluence photoacoustic imaging," Biomed. Opt. Express **11**(6), 3360–3373 (2020).
[23]E. M. A. Anas *et al.*, "Enabling fast and high quality LED photoacoustic imaging: A recurrent neural networks based approach," Biomed. Opt. Express **9**(8), 3852–3866 (2018).
[24]M. Kuniyil Ajith Singh *et al.*, "Deep learning-enhanced LED-based photoacoustic imaging," Proc. SPIE **11240**, 1124038 (2020).
[25]A. Paul and S. Mallidi, "U-Net enhanced real-time LED-based photoacoustic imaging," J. Biophot. **17**, e202300465 (2024).
[26]M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784 (2014).
[27]P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017).
[28]A. Creswell *et al.*, "Generative adversarial networks: An overview," IEEE Signal Process. Mag. **35**(1), 53–65 (2018).
[29]X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," Med. Image Anal. **58**, 101552 (2019).
[30]A. Paul and S. Mallidi, "Enhancement of signal-to-noise ratio for real-time LED-based photoacoustic imaging systems using denoising cycle-consistent generative adversarial networks," Proc. SPIE **12842**, 128420F (2024).
[31]A. F. Calvarons, "Improved Noise2Noise denoising with limited data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2021).
[32]J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," arXiv:1803.04189 (2018).
[33]A. Limsuebchuea, R. Duangsoithong, and P. Phukpattaranont, "Self-augmented noisy image for Noise2Noise image denoising," IEEE Access **12**, 71076–71087 (2024).
[34]Y. Mansour and R. Heckel, "Zero-shot Noise2Noise: Efficient image denoising without any data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2023).
[35]H. Adams *et al.*, "Persistence images: A stable vector representation of persistent homology," J. Mach. Learn. Res. **18**(8), 1–35 (2017).
[36]V. Kovacev-Nikolic *et al.*, "Using persistent homology and dynamical distances to analyze protein binding," Stat. Appl. Genet. Mol. Biol. **15**(1), 19–38 (2016).
[37]G. Naitzat, A. Zhitnikov, and L.-H. Lim, "Topology of deep neural networks," J. Mach. Learn. Res. **21**(184), 1–40 (2020).
[38]A. Paul and S. Mallidi, "Enhancing signal-to-noise ratio in real-time LED-based photoacoustic imaging: A comparative study of CNN-based deep learning architectures," Photoacoustics **41**, 100674 (2025).
[39]J. Henry, T. Natalie, and D. Madsen, *Pix2Pix Gan for Image-To-Image Translation* (Research Gate Publication, 2021), pp. 1–5.
[40]Y. Qu *et al.*, "Enhanced Pix2Pix dehazing network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019).
[41]P. Salehi and A. Chalechale, "Pix2Pix-based stain-to-stain translation: A solution for robust stain normalization in histopathology images analysis," in *2020 International Conference on Machine Vision and Image Processing (MVIP)* (IEEE, 2020).
[42]J. Sun *et al.*, "Pix2Pix generative adversarial network for low dose myocardial perfusion SPECT denoising," Quant. Imaging Med. Surg. **12**(7), 3539 (2022).
[43]I. A. Ibragimov and R. Z. Khas'minskii, "On nonparametric estimation of the value of a linear functional in Gaussian white noise," Theory Probab. Its Appl. **29**(1), 18–32 (1985).
[44]X. Zhang *et al.*, "A novel image encryption scheme based on CCNN," Physica Scr. **99**(2), 025253 (2024).
[45]M. Jia and M. Dong, "Analysis and comparison of Gaussian noise denoising algorithms," in *Journal of Physics: Conference Series* (IOP Publishing, 2021).
[46]S. Suryanarayana *et al.*, "Estimation and removal of Gaussian noise in digital images," Int. J. Electron. Commun. Eng. **5**(1), 23–33 (2012).
[47]S. W. Hasinoff, "Photon, poisson noise," Comput. Vision 608–610 (2016).
[48]J. Azzeh, B. Zahran, and Z. Alqadi, "Salt and pepper noise: Effects and removal," JOIV: Int. J. Inf. Visualization **2**(4), 252–256 (2018).
[49]M. Tur, K.-C. Chin, and J. W. Goodman, "When is speckle noise multiplicative?," Appl. Opt. **21**(7), 1157–1159 (1982).
[50]N. Yahya, N. S. Kamel, and A. S. Malik, "Subspace-based technique for speckle noise reduction in ultrasound images," Biomed. Eng. Online **13**(1), 154 (2014).
[51]W. H. Press and S. A. Teukolsky, "Savitzky–Golay smoothing filters," Comput. Phys. **4**(6), 669–672 (1990).
[52]J. S. Lim, *Two-Dimensional Signal and Image Processing* (Prentice Hall, Englewood Cliffs, 1990).
[53]T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," SIAM J. Sci. Comput. **20**(6), 1964–1977 (1999).
[54]A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (IEEE, 2005).
[55]K. Dabov *et al.*, "Image denoising with block-matching and 3D filtering," Proc. SPIE **6064**, 606414 (2006).
[56]O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2015).
[57]K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," arXiv:1702.05659 (2017).
[58]J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019).
[59]H. Zhao *et al.*, "Loss functions for neural networks for image processing," arXiv:1511.08861 (2015).
[60]A. Abu-Srhan, M. A. M. Abushariah, and O. S. Al-Kadi, "The effect of loss function on conditional generative adversarial networks," J. King Saud Univ. Comput. Inf. Sci. **34**(9), 6977–6988 (2022).
[61]K. Kurach *et al.*, "A Large-Scale Study on Regularization and Normalization in GANs," International Conference on Machine Learning **97**, (2019).
[62]L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks* (Springer Nature, 2012).
[63]J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res. **12**, 2121 (2011).

[64]Y. Dauphin *et al.*, "RMSProp and equilibrated adaptive learning rates for non-convex optimization," arXiv:1502.04390v2 (2015).
[65]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).
[66]Y. Nesterov, "A method for solving the convex programming problem with convergence rate O(1/k^2)," Proc. USSR Acad. Sci. **269**, 543–547 (1983).
[67]S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:abs/1609.04747 (2016).
[68]L. Liu *et al.*, "On the variance of the adaptive learning rate and beyond," arXiv:abs/1908.03265 (2019).
[69]J. Zhuang *et al.*, "AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients," arXiv:abs/2010.07468 (2020).
[70]M. Zaheer *et al.*, "Adaptive methods for nonconvex optimization," in *Neural Information Processing Systems* (NIPS, 2018).
[71]A. K. Dubey and V. Jain, "Comparative study of convolution neural network's Relu and Leaky-Relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018* (Springer, 2019).
[72]Z. Zhang, M. Li, and J. Yu, "On the convergence and mode collapse of GAN," in *SIGGRAPH Asia 2018 Technical Briefs* (ACM DL, 2018), pp. 1–4.
[73]S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 (ACM DL, 2015).
[74]T. Salimans and D. P. Kingma, *Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks* (NIPS, 2016).
[75]T. Miyato *et al.*, "Spectral normalization for generative adversarial networks," arXiv:abs/1802.05957 (2018).
[76]A. Zomorodian, "Fast construction of the Vietoris–Rips complex," Comput. Graphics **34**(3), 263–271 (2010).
[77]M. Adamaszek and H. Adams, "The Vietoris–Rips complexes of a circle," Pac. J. Math. **290**(1), 1–40 (2017).
[78]C. Tralie, N. Saul, and R. Bar-On, "Ripser.py: A lean persistent homology library for Python," J. Open Source Softw. **3**(29), 925 (2018).
[79]Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," Telecommun. Syst. **49**, 35–48 (2012).
[80]Z. Wang *et al.*, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process. **13**, 600–612 (2004).
[81]V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *International Conference on Machine Learning* (ACM DL, 2010).
[82]B. Xu *et al.*, "Empirical evaluation of rectified activations in convolutional network," arXiv:abs/1505.00853 (2015).
[83]K. Bahrami and A. C. Kot, "A fast approach for no-reference image sharpness assessment based on maximum local variation," IEEE Signal Process. Lett. **21**, 751–755 (2014).