

A benchmark dataset for quantum noise effects in amplitude-encoded images

Ola Al-Ta'ani

olataani@yu.edu.jo

Yarmouk University

Data Note

Keywords: Quantum image processing, Amplitude encoding, Quantum noise, Benchmark dataset, Amplitude damping, Dephasing, Depolarizing noise, Quantum trajectories, Image denoising, BSDS500

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11381515/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

A benchmark dataset for quantum noise effects in amplitude-encoded images

Ola Al-Ta'ani*

Department of Computer Engineering, Yarmouk University, Irbid, Jordan

*Corresponding author: Ola Al-Ta'ani (olataani@yu.edu.jo)

Abstract

Quantum image processing methods that rely on amplitude encoding are sensitive to channel-induced noise, yet reusable benchmarks for evaluating these effects under controlled conditions remain limited. We present QN-AEI, a simulation-based benchmark constructed from 200 images in the BSDS500 training subset. Each image is resized to 64×64 pixels and normalized to a 4096-dimensional amplitude vector representing a 12-qubit quantum state, then subjected to three noise channels—amplitude damping, Pauli-Z dephasing, and depolarizing noise—at four levels, $\eta \in \{0.1, 0.3, 0.5, 0.7\}$, with 200 stochastic trajectory realizations per image-channel-noise condition. This yields 2,400 conditions, 480,000 noisy realizations, and 1,966,080,000 normalized perturbation samples in total. Each record pairs the clean image and amplitude vector with its noisy statevectors, raw and normalized perturbations, and condition-level metadata; the release also includes source-image provenance, a dataset manifest, validation outputs, summary statistics, documentation, and the MATLAB generation and validation code. QN-AEI supports reproducible benchmarking of quantum-noise characterization, denoising, uncertainty and reliability analysis, and bounded-domain perturbation modeling.

Background & Summary

Amplitude encoding maps a normalized classical vector into the amplitudes of a quantum state and is an important representation in quantum image processing and quantum machine learning [2]. In image-based settings, a grayscale image can be vectorized and normalized so that its pixel intensities define the amplitudes of a quantum state, one of several quantum image representation schemes proposed in the literature, including FRQI [3]

and NEQR [4] (see [7] for a broader survey). This representation is attractive because it embeds high-dimensional image information into a compact quantum-state form. At the same time, it makes the encoded data sensitive to imperfections in state preparation, quantum evolution, and noisy simulation or measurement processes [2, 5]. Decoherence, gate errors, and channel-dependent perturbations can alter the encoded state in ways that degrade image fidelity and distort amplitude-level information. Controlled noise benchmarks are therefore important for developing, evaluating, and comparing quantum image processing and quantum machine learning methods under reproducible noise conditions [13].

Recent work has used standard image datasets and benchmark images to investigate quantum image loading, amplitude embedding, and related quantum-data representations in simulated and hardware-aware settings [5, 6]. A separate line of work has benchmarked the robustness of quantum machine learning models to adversarial perturbations and hardware-level noise on standard image datasets [8, 9], but these efforts evaluate model-level robustness rather than releasing a systematically perturbed, matched clean/noisy image benchmark. Dedicated quantum dataset resources, such as QDataSet [10], provide controlled quantum-state records for machine learning tasks, but they do not specifically address image-based amplitude encoding or channel-specific noise effects on visual data. More broadly, existing resources in quantum image processing and quantum data benchmarking generally emphasize representation models, loading procedures, and algorithm development, rather than providing a structured benchmark with matched clean inputs, controlled channel-specific noisy outputs, repeated stochastic realizations, perturbation records, and reuse-oriented metadata. To the best of our knowledge, publicly available datasets designed specifically to document simulated quantum noise effects in amplitude-encoded natural images remain limited.

The design of QN-AEI was chosen to balance visual diversity, controlled quantum-noise variation, repeated stochastic sampling, and practical dataset size. The 200-image BSDS500 training subset was used because it provides a fixed and widely recognized collection of natural images with diverse edges, textures, smooth regions, object structures, and intensity distributions, while avoiding arbitrary post-hoc image selection. The fixed 64×64 resolution yields amplitude vectors of length $4096 = 2^{12}$, enabling a consistent 12-qubit representation for all records and direct comparison across images, channels, and noise levels.

The four noise levels $\eta \in \{0.1, 0.3, 0.5, 0.7\}$ provide four representative channel-parameter values spanning $\eta = 0.1$ to 0.7 without creating an un-

manageably large release. This grid is not intended to exhaust the continuous range of possible channel parameters; rather, it provides a controlled set of representative operating points for benchmarking, statistical modeling, denoising, and reliability analysis. For amplitude damping and depolarizing noise, η is the per-qubit relaxation or error probability and larger values correspond to stronger noise. For the Pauli-Z dephasing channel of Eq. (5) the correspondence is less direct, because the coherence factor of the ensemble-averaged channel is $1 - 2\eta$: the magnitude of this coherence factor decreases to zero at $\eta = 0.5$ and increases again, with sign reversal, for $\eta > 0.5$. QN-AEI stores individual quantum trajectories rather than ensemble-averaged density matrices, and the aggregated mean normalized perturbation reported in Table 5 increases with η for all three channels. The parameter η should therefore be read as a channel parameter rather than as a monotone severity scale for PD. Similarly, $R = 200$ repeated realizations per image-channel-noise condition were used to provide stochastic replication for estimating condition-level perturbation summaries and realization-dependent variability, while keeping the dataset computationally and storage-wise practical for reuse.

The Quantum Noise Effects in Amplitude-Encoded Images dataset (QN-AEI) was created to address this need, providing a curated simulation-based benchmark for studying quantum noise effects in amplitude-encoded grayscale images under controlled and reproducible conditions. It is built from the same 200-image BSDS500 training subset, resolution, amplitude-encoding scheme, and channel/noise-level design introduced above [1]; the full construction procedure, noise-channel definitions, and resulting record counts are given in Methods and Data Records, below.

Each QN-AEI image-level record pairs a clean image with its noisy realizations and perturbation data under structured condition-level metadata, as summarized above; the full record schema is given in Data Records, below. The release is accompanied by the MATLAB generation and validation script, source-image provenance information, a dataset manifest, validation outputs, and quantitative summary statistics, intended together to make the generation workflow transparent and to support reproducible reuse of the released records.

The release was organized with FAIR-oriented reuse in mind [11, 12]. The repository record, persistent identifier, version information, citation details, structured metadata, and the MD5 checksum that the repository computes and publishes automatically for every uploaded file support discovery and integrity checking. CSV provenance files, documented variable schemas, and structured MATLAB `-v7.3` records facilitate machine-readable reuse, while

source-image provenance, documented preprocessing and simulation settings, deterministic condition-level seeds, validation outputs, documented loading examples, and the released MATLAB generation and validation code support reproducibility and extension of the benchmark.

Because the source images are derived from the BSDS500 training subset [1], the release documents the source-image terms and distinguishes BSDS500-derived image components from the newly generated QN-AEI records, noisy realizations, perturbation arrays, metadata, validation outputs, documentation, and the MATLAB script. Users should cite the original BSDS500 reference when reusing or extending the dataset.

QN-AEI is intended to support several complementary lines of quantum-noise-modeling research, including benchmarking, denoising, restoration, and noise-mitigation strategies for amplitude-encoded image states [14, 15, 16], statistical characterization of bounded-domain perturbations, and reliability analysis under repeated realizations; these reuse scenarios are discussed further in Usage Notes, below.

The present article provides a dedicated description of the released QN-AEI benchmark and focuses on documenting the dataset design, generation workflow, data records, validation procedures, provenance, and intended reuse.

Methods

Source images, preprocessing, and amplitude encoding

QN-AEI was constructed from the 200 images in the training subset of the Berkeley Segmentation Dataset and Benchmark 500 (BSDS500) [1]. This subset was used as a fixed and well-established natural-image collection, enabling reproducible generation of image-derived amplitude-encoded records without arbitrary post-hoc image selection. The original BSDS500 image identifiers, source subset, original image dimensions, color type, applied preprocessing steps, and corresponding QN-AEI record filenames are documented in the released provenance file `source_image_provenance.csv`. This provenance file provides the primary reference for linking each generated QN-AEI record to its source BSDS500 image. Representative examples from this subset are shown in Fig. 1.

Each source image was converted to grayscale when needed using `rgb2gray`, resized to 64×64 pixels using bicubic interpolation with `imresize`, converted to double precision using `im2double`, and scaled to the



Figure 1: Representative source images from the BSDS500 training subset used to construct QN-AEI. The examples illustrate the diversity of visual structures, textures, intensity distributions, and content types present in the 200-image collection. Image labels correspond to the original BSDS500 image identifiers. The complete source-image provenance information, including the original BSDS500 image identifiers and corresponding QN-AEI record filenames, is provided in `source_image_provenance.csv`.

interval $[0, 1]$. The resulting preprocessed image is denoted by

$$I \in [0, 1]^{64 \times 64}.$$

Each image was then vectorized in MATLAB column-major order as

$$a = \text{vec}(I) \in \mathbb{R}^{4096}, \quad (1)$$

where the fixed resolution of 64×64 pixels ensures that all amplitude vectors have length $4096 = 2^{12}$. This common vector length supports consistent amplitude encoding and direct comparison across all released image-channel-noise configurations.

To construct a valid amplitude-encoded quantum state representation, each vectorized image was normalized to unit ℓ_2 norm. Since the selected BSDS500 images contain nonzero grayscale intensity structure after preprocessing, the ℓ_2 norm of each vectorized image is positive. The clean amplitude

vector was computed as

$$\psi = \frac{a}{\|a\|_2} \in \mathbb{R}^{4096}, \quad (2)$$

so that the unit-norm constraint required for a valid quantum statevector is satisfied:

$$\sum_{i=1}^{4096} |\psi_i|^2 = 1. \quad (3)$$

Because $4096 = 2^{12}$, each preprocessed image corresponds to a 12-qubit amplitude-encoded quantum state [2].

The clean amplitude vector ψ is real-valued by construction, directly reflecting the non-negative grayscale intensities of the preprocessed source image after normalization. The application of quantum noise channels, described in the following subsection, may introduce sign changes or complex phases into the noisy statevectors; the resulting noisy realizations are therefore stored as vectors in $\mathbb{C}^{4096}$. In the released image-level records, the clean amplitude vector is stored as `a_clean`.

Quantum-noise simulation

For each amplitude-encoded image, noisy realizations were generated under three standard completely positive trace-preserving (CPTP) quantum noise channels: amplitude damping (AD), Pauli- Z dephasing (PD), and depolarizing noise (DP) [21, 17]. The label PD is retained for the dephasing channel throughout the release for compatibility with the stored record fields.

Channel-selection rationale. The three channels were selected to represent complementary and widely used classes of quantum noise affecting amplitude-encoded states. Amplitude damping represents non-unitary energy relaxation and directly changes amplitude magnitudes. Pauli- Z dephasing represents phase and coherence perturbations that can alter the signed or complex amplitude vector while leaving reconstructed magnitude images comparatively close to the clean reference. Depolarizing noise represents symmetric stochastic Pauli errors and provides a stronger generic corruption model that can affect both amplitude magnitudes and phases. Together, these channels provide a compact but diverse benchmark covering relaxation-dominated, phase-dominated, and Pauli-error-dominated noise behavior.

The selected channels are not intended to exhaust all physically relevant noise models. Other channels, such as bit-flip noise, generalized amplitude

damping, correlated noise, or hardware-calibrated device noise, are important extensions. They were not included in the present release in order to maintain a controlled, interpretable, and reusable benchmark design with fixed images, fixed qubit count, matched channel/noise configurations, and repeated realizations.

Each channel was evaluated at four noise levels,

$$\eta \in \{0.1, 0.3, 0.5, 0.7\},$$

with $R = 200$ stochastic noisy realizations generated for every image–channel–noise configuration. Noisy statevector realizations were produced using a Monte Carlo quantum-trajectory framework [19, 20] applied sequentially to the 12 qubits, without explicitly storing or evolving full density matrices. For a clean amplitude vector ψ , the corresponding noisy realization is denoted by ψ' .

For completeness, the trajectory simulation can be viewed as an unraveling of the corresponding density-matrix channel [19, 20]. For a channel with Kraus operators $\{E_k\}$ and a pure input state $\rho = |\psi\rangle\langle\psi|$, trajectory branch k is selected with probability

$$p_k = \langle\psi|E_k^\dagger E_k|\psi\rangle = \|E_k|\psi\rangle\|_2^2,$$

and the corresponding normalized trajectory state is

$$|\psi'_k\rangle = \frac{E_k|\psi\rangle}{\sqrt{p_k}}.$$

Averaging over trajectory outcomes gives

$$\mathbb{E} [|\psi'_k\rangle\langle\psi'_k|] = \sum_k p_k \frac{E_k|\psi\rangle\langle\psi|E_k^\dagger}{p_k} = \sum_k E_k \rho E_k^\dagger,$$

which is exactly the CPTP map. In QN-AEI, this sampling procedure is applied sequentially across the 12 qubits to generate one stochastic noisy statevector realization.

To ensure reproducibility, a base random seed of 1 was used. Rather than relying on a single global random stream, QN-AEI uses deterministic condition-level seeds generated from the base seed, the source-image identifier, the noise channel, and the noise level. This design makes each image–channel–noise configuration reproducible independently of the number or order of images processed. The resulting condition-level seed is stored in each released condition record as `condition_seed`.

Qubit indexing follows a most-significant-qubit convention. Specifically, qubit 1 acts on the most significant computational-basis index of the amplitude vector, whose entries are ordered as

$$|q_1 q_2 \cdots q_{12}\rangle,$$

with q_1 representing the most significant binary index. This convention is used consistently when applying all single-qubit channel operators.

This convention does not change the abstract quantum channel applied to the 12-qubit state, but it determines how single-qubit trajectory operations map onto the linearized amplitude vector and, after reshaping, onto spatial image locations. The convention is therefore documented explicitly so that users can reproduce the same reconstructed-image effects and spatial perturbation patterns. Alternative bit-ordering conventions would represent a different mapping between qubit indices and image-grid locations and should not be mixed with the released records without explicit conversion.

Amplitude damping

Amplitude damping models irreversible energy relaxation from the excited state $|1\rangle$ to the ground state $|0\rangle$. The corresponding CPTP map is defined by the Kraus operators

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\eta} \end{pmatrix}, \quad E_1 = \begin{pmatrix} 0 & \sqrt{\eta} \\ 0 & 0 \end{pmatrix}, \quad (4)$$

where $\eta \in [0, 1]$ is the relaxation probability. In the Monte Carlo trajectory implementation, the operators E_0 and E_1 are applied to each qubit according to their state-dependent branch probabilities, followed by renormalization of the resulting statevector. Applying this procedure across all 12 qubits yields one stochastic realization of the amplitude-damped state. This channel directly alters amplitude magnitudes and can therefore produce visible structural distortions in reconstructed magnitude images.

Pauli-Z dephasing

Pauli-Z dephasing models decoherence that suppresses relative phase coherence while preserving population probabilities. It is implemented as a Pauli-Z trajectory acting on each qubit independently:

$$\rho \mapsto (1 - \eta) \rho + \eta Z \rho Z, \quad (5)$$

where ρ is the qubit density matrix and $\eta \in [0, 1]$ is the probability of applying the Pauli- Z operation. In the trajectory implementation, each qubit independently receives a Pauli- Z operation with probability η or the identity operation with probability $1 - \eta$. Because the Pauli- Z operation changes relative signs/phases while preserving component magnitudes, reconstructed magnitude images can remain visually close to the clean reference. Nevertheless, the signed amplitude vector is changed, and these changes are captured by the stored perturbation records.

Depolarizing noise

Depolarizing noise models stochastic Pauli errors. The channel acts on each qubit as

$$\rho \mapsto (1 - \eta) \rho + \frac{\eta}{3} (X \rho X + Y \rho Y + Z \rho Z), \quad (6)$$

where $\eta \in [0, 1]$ controls the error rate. In the trajectory implementation, each qubit is left unchanged with probability $1 - \eta$ or receives one of the Pauli operators X , Y , or Z , each with probability $\eta/3$. This channel perturbs both amplitude magnitudes and phases and often produces spatially irregular distortions in reconstructed magnitude images.

Global phase alignment and renormalization

Before perturbations were computed, each noisy statevector was aligned to the clean reference by removing the global phase ambiguity. The aligned noisy state was computed as

$$\psi' \leftarrow \psi' \cdot e^{-i \angle(\psi'^{\dagger} \psi')}, \quad (7)$$

where $\angle(\cdot)$ denotes the argument of a complex scalar. This step ensures that componentwise differences reflect amplitude-level deviations rather than an arbitrary global phase offset. Each noisy statevector was also checked for unit norm after channel application; statevectors deviating from unit norm by more than 10^{-10} were renormalized before perturbation extraction.

The same matched design was applied to all images, channels, and noise levels, yielding

$$200 \times 3 \times 4 = 2,400$$

image-channel-noise configurations and

$$2,400 \times 200 = 480,000$$

noisy amplitude-encoded realizations in the released dataset. Representative examples of channel-dependent effects on reconstructed magnitude images are shown in Fig. 2.

Perturbation construction

For each noisy realization, perturbations were computed by comparing the clean and noisy amplitudes componentwise after global phase alignment as described above. Let ψ_i and ψ'_i denote the i th components of the clean and aligned noisy statevectors, respectively. The raw perturbation magnitude was computed as

$$\Delta_i = |\psi'_i - \psi_i|, \quad i = 1, \dots, 4096. \quad (8)$$

The normalized perturbation value stored for bounded-domain analysis was defined as

$$y_i = \frac{\Delta_i}{2}, \quad 0 \leq y_i \leq 1. \quad (9)$$

The division by 2 follows from the theoretical upper bound $\Delta_i \leq 2$, which holds for all unit-norm statevectors. Since both ψ and ψ' satisfy $\|\psi\|_2 = \|\psi'\|_2 = 1$, each component is bounded in magnitude by unity, and the triangle inequality gives

$$\Delta_i = |\psi'_i - \psi_i| \leq |\psi'_i| + |\psi_i| \leq 1 + 1 = 2. \quad (10)$$

This bound holds regardless of the noise channel or noise level. As an additional empirical check, the generation script raises a hard error if any value of Δ_i exceeds $2 + 10^{-10}$ during dataset production; no such violation was observed across the released perturbation records.

The normalization by the theoretical upper bound of 2 was chosen deliberately rather than normalizing by the empirical maximum observed in the dataset. A data-dependent maximum would vary across channels, noise levels, images, or future dataset extensions, making perturbation values less comparable across conditions. In contrast, the bound $\Delta_i \leq 2$ is channel-independent, image-independent, and dimension-independent for unit-norm statevectors. The resulting normalized value $y_i = \Delta_i/2$ therefore provides a common absolute perturbation scale for all released records.

The observed perturbations do not necessarily occupy the full interval $(0, 1)$. In the present release, the largest observed normalized perturbation was below 0.5, indicating that the simulated noise conditions occupy a physically and empirically smaller subregion of the theoretical componentwise range.

This does not affect comparability, because the raw perturbations Δ_i are also stored and the normalized values preserve the same ordering and relative differences under a fixed global scale. Users who require condition-specific rescaling can derive alternative normalized variables from the released raw perturbation arrays.

To support numerical stability in subsequent bounded-domain distribution analysis, each normalized perturbation value was clipped to the open interval $(0, 1)$ by replacing values below $\varepsilon = 10^{-12}$ with ε and values above $1 - \varepsilon$ with $1 - \varepsilon$. This operation changes only values at or extremely close to the boundaries, and preserves the bounded-domain structure required by the released statistical summaries; the unclipped raw perturbations remain available in `delta_raw` for users who need them.

For each image-channel-noise configuration, perturbation values from all 4096 amplitude components and $R = 200$ realizations were pooled into a single condition-level vector, producing

$$4096 \times 200 = 819,200$$

normalized perturbation samples per condition. Across the full 200-image dataset, this gives

$$200 \times 3 \times 4 \times 4096 \times 200 = 1,966,080,000$$

normalized perturbation samples in total.

Record generation and storage

The final released collection contains the preprocessed clean images, their amplitude-encoded representations, noisy statevector realizations, raw perturbation matrices, pooled perturbation arrays, normalized perturbation samples, and structured condition-level metadata. The main image-level records follow the naming convention

$$\text{qnaei_bsd_record_}<\text{BSDS500-image-id}>_R200.\text{mat}$$

where `<BSDS500-image-id>` is the original BSDS500 source-image identifier without extension. For example, the source image `100075.jpg` is stored as `qnaei_bsd_record_100075_R200.mat`.

Each image-level record is a MATLAB `.mat` file containing the image-level variables summarized in Table 2. At the top level, each record stores the preprocessed clean image `I`, the normalized clean amplitude vector `a_clean`, the image filename `img_name`, the full source-image list `img_list`, the image

dimensions `img_size`, the number of amplitude components `N`, the number of qubits `n_qubits`, the base random seed `rng_seed`, a generation timestamp `run_timestamp`, a code version string `code_version`, the source-dataset label `source_dataset`, and the documented reuse terms `usage_terms`. The condition-level data are stored in a 3×4 cell array `cond_data`, where rows correspond to the three noise channels (AD, PD, DP) and columns correspond to the four noise levels $\eta \in \{0.1, 0.3, 0.5, 0.7\}$.

Each cell of `cond_data` contains a structure with the condition-level fields summarized in Table 3. These include the noisy statevector matrix `psi_noisy` of size 4096×200 , the raw perturbation matrix `delta_raw` of the same size, the pooled perturbation vector `delta_pool` of length 819,200, and the normalized and clipped perturbation vector `y` of the same length. Each condition structure also stores the channel label, noise level, realization count, deterministic condition-level seed `condition_seed`, normalization rule, global-phase removal flag, and numerical clipping threshold.

To support dataset-level navigation and reproducibility, the release also includes a manifest file, `qnaei_manifest_R200_200img_BSD.mat`, that records the complete image list, generation settings, image size, channels, noise levels, realization count, base random seed, run timestamp, code version, output directory, source-dataset label, reuse terms, and provenance-file location. Source-image provenance is additionally documented in `source_image_provenance.csv`, which links each QN-AEI record to its BSDS500 source image and records the original image identifier, source subset, dimensions, color type, and applied preprocessing steps.

As described above, records were generated using the deterministic condition-level seeding scheme, ensuring that every image-channel-noise configuration remains reproducible independently of processing order. The overall workflow consists of source-image preprocessing, amplitude encoding, quantum-noise simulation, perturbation construction, structured record generation, and automated validation. The released MATLAB script reproduces each step and provides users with the code required to inspect, regenerate, and validate the released records.

Data Records

The QN-AEI dataset is organized as a structured benchmark resource containing source-image provenance information, image-level MATLAB records, metadata, documentation, validation outputs, summary statistics, and the MATLAB generation and validation script. The dataset is deposited in

Zenodo under the persistent identifier *doi:10.5281/zenodo.22697780*. The released version contains 200 image-level records, one for each image from the BSDS500 training subset used in the dataset. Each record stores the preprocessed clean image, its amplitude-encoded representation, noisy amplitude-state realizations generated under the full set of noise conditions, perturbation arrays, and condition-level metadata.

At the top level, the released package contains a README file and a license file together with folders for image-level records, metadata, code, validation outputs, and results. The image-record folder contains the main MATLAB records, with one file per source image following the naming convention

`qnaei_bsd_record_<BSDS500-image-id>_R200.mat`

For example, the record associated with the BSDS500 source image 100075.jpg is stored as

`qnaei_bsd_record_100075_R200.mat`

Each image-level record contains the preprocessed clean image, the corresponding normalized amplitude vector, and a structured condition array for the three quantum noise channels and four noise levels. The condition array has size 3×4 , corresponding to amplitude damping, Pauli-Z dephasing, and depolarizing noise across $\eta \in \{0.1, 0.3, 0.5, 0.7\}$. For each image-channel-noise condition, the record stores $R = 200$ noisy realizations. Thus, the released dataset contains

$$200 \times 3 \times 4 = 2,400$$

image-channel-noise conditions and

$$2,400 \times 200 = 480,000$$

noisy amplitude-encoded realizations in total.

The perturbation data are stored within the image-level records. For each condition, the noisy state matrix has size 4096×200 , where rows correspond to amplitude components and columns correspond to repeated noisy realizations. The associated raw perturbation matrix has the same size. The pooled raw perturbation vector contains

$$4096 \times 200 = 819,200$$

samples per condition, and the normalized perturbation vector stores the corresponding bounded-domain values. Across the complete 200-image release, the dataset contains

$$200 \times 3 \times 4 \times 4096 \times 200 = 1,966,080,000$$

normalized perturbation samples.

The metadata folder contains the source-image provenance file linking each record to its BSDS500 source image. The code folder contains the MATLAB script implementing preprocessing, amplitude encoding, quantum-noise simulation, perturbation construction, record generation, and validation. The validation and results folders contain the automated validation report and the quantitative perturbation summary statistics. File integrity is verified through the MD5 checksum that the repository computes and publishes automatically for every uploaded file, rather than through a separately distributed checksums file. The README and manifest files serve as the main entry points for users and describe how the folders, records, metadata, validation outputs, and results should be interpreted together.

Table 1 summarizes the principal file groups included in the release. Tables 2 and 3 summarize the image-level and condition-level variables stored in each MATLAB record.

Table 1: Main file groups in the QN-AEI dataset release.

File group	Format	Description
Source-image provenance	CSV	Metadata linking each QN-AEI record to its BSDS500 source image, including image identifier, source subset, original dimensions, color type, usage terms, preprocessing steps, and record filename. The original BSDS500 images are not redistributed as standalone image files in the release.
Image-level records	MAT	One record per BSDS500 source image containing the clean image, clean amplitude vector, noisy state matrices, perturbation arrays, and condition metadata.
Manifest	MAT	Single dataset-level manifest recording the shared generation configuration: image list, channels, noise levels, realization count, base random seed, run timestamp, code version, and provenance-file location.
Generation and validation script	M	Single MATLAB script implementing preprocessing, amplitude encoding, quantum-noise simulation, perturbation construction, record generation, and automated validation.
Validation outputs	CSV	Automated validation report documenting structural checks, completeness checks, dimensional consistency, numerical bounds, and file readability for every released record.
Results / statistics	CSV	Quantitative perturbation summaries aggregated by noise channel and noise level, together with condition-level perturbation statistics.
README / license	MD / TXT	Top-level documentation describing file relationships, package contents, generation settings, dataset navigation, and license terms.

Technical Validation

The technical quality of QN-AEI was assessed through automated consistency, completeness, dimensionality, bounded-domain, file-readability, and empirical distributional checks applied to the generated records. These checks were designed to verify that the released files conform to the documented dataset schema, that all image-channel-noise configurations are complete and internally consistent, and that the stored perturbation values exhibit interpretable channel- and noise-level-dependent behavior. The dataset statistics

Table 2: Image-level variables stored in each QN-AEI MATLAB record (`qnaei_bsd_record_<BSDS500-image-id>_R200.mat`).

Variable	Size / type	Description
<code>I</code>	64×64 double	Preprocessed grayscale image scaled to $[0, 1]$.
<code>a_clean</code>	4096×1 double	Normalized clean amplitude vector with unit ℓ_2 norm.
<code>img_name</code>	character array / string	Source image filename including extension, e.g., <code>100075.jpg</code> .
<code>img_list</code>	1×200 cell	Complete list of the 200 BSDS500 source image filenames used in the release.
<code>img_size</code>	1×2 double	Image dimensions, <code>[64 64]</code> .
<code>N</code>	scalar integer	Amplitude vector length, 4096.
<code>n_qubits</code>	scalar integer	Number of qubits, 12.
<code>channels</code>	1×3 cell	Noise channel labels: <code>{'AD', 'PD', 'DP'}</code> .
<code>eta_levels</code>	1×4 double	Noise levels: <code>[0.1 0.3 0.5 0.7]</code> .
<code>R</code>	scalar integer	Number of noisy realizations per condition, 200.
<code>rng_seed</code>	scalar integer	Base random seed used for deterministic condition-level seeding, 1.
<code>run_timestamp</code>	datetime	Generation timestamp.
<code>code_version</code>	character array / string	Dataset-generation version identifier.
<code>cond_data</code>	3×4 cell array	Condition-level records for all channel-noise combinations.
<code>epsv</code>	scalar double	Numerical clipping threshold, 10^{-12} .
<code>norm_tol</code>	scalar double	Unit-norm tolerance used in validation and normalization checks.
<code>range_tol</code>	scalar double	Numerical tolerance used for perturbation range checks.
<code>remove_global_phase</code>	logical	Flag indicating whether global phase alignment was applied.
<code>source_dataset</code>	character array / string	Source dataset label, BSDS500 training subset.
<code>usage_terms</code>	character array / string	Documented reuse terms associated with the source-image-derived release.

Table 3: Condition-level fields stored within each cell of `cond_data`. Each cell corresponds to one image–channel–noise configuration.

Field	Size / type	Description
<code>channel</code>	character array / string	Noise channel label: 'AD', 'PD', or 'DP'.
<code>eta</code>	scalar double	Noise level η for this condition.
<code>R</code>	scalar integer	Number of noisy realizations, 200.
<code>condition_seed</code>	scalar integer	Deterministic seed for this image–channel–noise condition.
<code>psi_noisy</code>	4096×200 complex	Matrix of noisy statevector realizations; columns correspond to repeated stochastic realizations. Stored as a real-valued array for AD conditions; see Usage Notes.
<code>delta_raw</code>	4096×200 double	Raw componentwise perturbation magnitudes Δ_i .
<code>delta_pool</code>	819200×1 double	Pooled raw perturbations across all amplitude components and realizations.
<code>y</code>	819200×1 double	Normalized and clipped perturbation samples $y_i = \Delta_i/2$.
<code>y_is_normalized</code>	logical	Flag confirming normalization, <code>true</code> .
<code>normalization_rule</code>	character array / string	Normalization formula: ' <code>y = delta_pool / 2</code> '.
<code>global_phase_removed</code>	logical	Flag confirming global phase alignment, <code>true</code> .
<code>epsv</code>	scalar double	Clipping threshold, 10^{-12} .

confirmed by these checks are summarized in Table 4.

Table 4: Summary of the released QN-AEI dataset.

Item	Value
Source images	200
Image source	BSDS500 training subset
Source-image provenance	<code>source_image_provenance.csv</code>
Image size	64×64 pixels
Amplitude vector length	4096
Amplitude-encoded qubits	12
Noise channels	AD, PD, DP
Noise levels η	0.1, 0.3, 0.5, 0.7
Realizations per condition	200
Conditions per image	12
Total conditions	2,400
Total noisy realizations	480,000
Perturbation samples per condition	819,200
Total perturbation samples	1,966,080,000
Base random seed	1
Seeding strategy	Deterministic condition-level seeds
Record format	MATLAB <code>.mat</code>

Automated structural and bounded-domain validation

Preprocessing checks. Each source image was converted to grayscale when needed, resized to 64×64 pixels, represented in double precision, and scaled to the interval $[0, 1]$. The vectorized image was then normalized to unit ℓ_2 norm before amplitude encoding. For each released record, validation confirmed that the stored clean image has size 64×64 , that the clean amplitude vector has length 4096, and that the amplitude vector satisfies the unit-norm condition $|\|\psi\|_2 - 1| \leq 10^{-10}$ required for valid amplitude encoding.

Completeness checks. Each of the 200 BSDS500 source images was required to contain records for the same three quantum noise channels (AD, PD, DP) and the same four noise levels $\eta \in \{0.1, 0.3, 0.5, 0.7\}$, with $R = 200$ noisy realizations for every image-channel-noise configuration. These checks confirmed that the released dataset contains

$$200 \times 3 \times 4 = 2,400$$

image-channel-noise configurations and

$$2,400 \times 200 = 480,000$$

noisy amplitude-encoded realizations.

Structural validation. Each image-level record was required to contain a 3×4 condition structure. For each condition, the noisy-state matrix was required to have size 4096×200 , the raw perturbation matrix was required to have the same size, and the pooled and normalized perturbation vectors were each required to contain

$$4096 \times 200 = 819,200$$

samples. Prior to perturbation extraction, each noisy statevector was verified to satisfy the unit-norm condition within tolerance 10^{-10} ; any state vector deviating beyond this threshold was renormalized before perturbations were computed. Each condition record was also checked for the presence of required metadata fields, including the channel label, noise level, realization count, deterministic condition seed, normalization rule, global-phase removal flag, and numerical clipping threshold.

Perturbation-domain checks. Raw perturbation values were verified to lie within the theoretical range $[0, 2]$, with a hard error raised during generation if any value exceeded $2 + 10^{-10}$. No such violation was observed across the released records, confirming that the normalization $y_i = \Delta_i/2$ maps all stored values to the bounded interval $[0, 1]$. After clipping to the open interval $(0, 1)$ using $\varepsilon = 10^{-12}$, the normalized perturbation values were also checked to remain within the documented numerical bounds. Across the full release, the dataset contains

$$200 \times 3 \times 4 \times 4096 \times 200 = 1,966,080,000$$

normalized perturbation samples. Metadata fields—including image identity, channel label, noise level, realization count, condition seed, and record naming—were cross-checked against the manifest and provenance files to ensure consistency between the stored data and the documented experimental design.

Visual channel-effect validation

Visual inspection was used as a complementary validation step to confirm that the simulated channels produced qualitatively distinct effects on reconstructed magnitude images. Figure 2 shows a representative BSDS500 example at $\eta = 0.1$. The figure is not intended as a complete characterization of the dataset, but as a visual check that the channel implementations produce interpretable and channel-specific behavior consistent with their definitions.

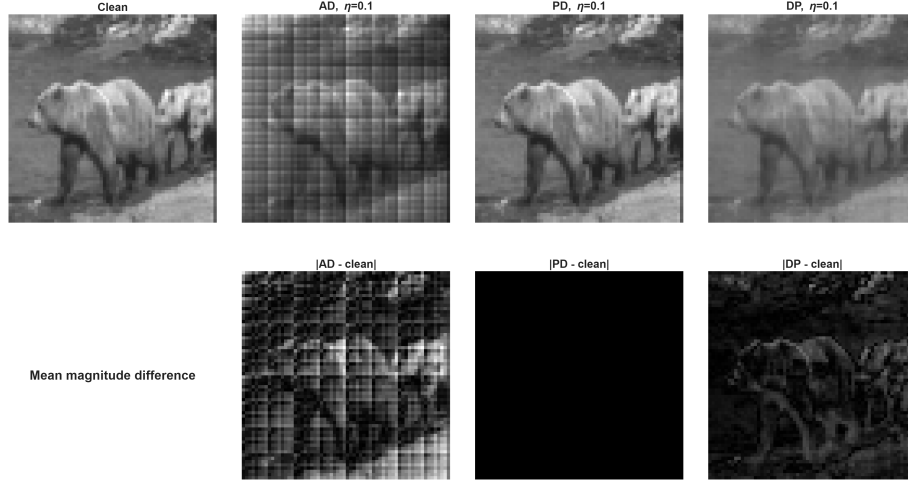


Figure 2: Representative channel effects on one BSDS500 source image at $\eta = 0.1$. Top row: clean magnitude image and mean reconstructed magnitude images over $R = 200$ noisy realizations for amplitude damping (AD), Pauli-Z dephasing (PD), and depolarizing noise (DP). Bottom row: corresponding mean magnitude-difference maps relative to the clean reference. Difference maps are displayed using a common intensity scale across channels. AD introduces structured attenuation patterns, PD preserves the magnitude image most strongly, and DP produces spatially irregular perturbations.

Quantitative perturbation characterization

In addition to structural and bounded-domain validation, the normalized perturbation samples were quantitatively characterized to verify that the simulated records exhibit systematic channel- and noise-level-dependent behavior. For each image-channel-noise configuration, summary statistics were computed from the condition-level vector of normalized perturbation samples. These statistics included the mean, standard deviation, median, selected upper quantiles, skewness, raw kurtosis, and excess kurtosis. The resulting condition-level summaries were then aggregated by noise channel and noise level across the 200 source images.

Table 5 summarizes the normalized perturbation distributions by channel and noise level. The statistics confirm that the perturbation records are not merely structurally complete, but also exhibit interpretable channel-specific behavior. For amplitude damping, the mean perturbation increases from 0.002999 at $\eta = 0.1$ to 0.007497 at $\eta = 0.7$, accompanied by a strong increase

in skewness and raw kurtosis. For Pauli-Z dephasing, the perturbation distribution remains comparatively concentrated, with near-zero median values at low noise levels and bounded upper quantiles across all noise levels. Depolarizing noise produces a clear upward shift in typical perturbation magnitude, with the median increasing from 0.001082 at $\eta = 0.1$ to 0.008425 at $\eta = 0.7$.

Across all channel-noise configurations, the minimum normalized perturbation value was 10^{-12} , reflecting the numerical clipping threshold, and the largest observed value was 0.48355 under amplitude damping at $\eta = 0.7$. These quantitative summaries complement the reconstructed-image examples in Fig. 2 and provide empirical evidence that QN-AEI captures distinguishable perturbation behavior across the three simulated noise channels and four noise levels. The full condition-level statistics and channel-level summary tables are included in the released validation outputs to support independent inspection and reuse.

Table 5: Quantitative summary of normalized perturbation samples by noise channel and noise level. Values report the mean of image-level condition summaries aggregated across the 200 BSDS500 source images. The kurtosis column reports raw kurtosis.

Channel	η	Mean	Std.	Median	Q90	Q99	Skew.	Kurt.
AD	0.1	0.002999	0.003733	0.001024	0.008736	0.013853	1.557	6.037
AD	0.3	0.006098	0.004619	0.005643	0.011090	0.021062	2.313	21.359
AD	0.5	0.007236	0.005600	0.006583	0.011555	0.028026	6.023	86.799
AD	0.7	0.007497	0.006736	0.006790	0.011390	0.025909	12.706	306.563
PD	0.1	0.005009	0.007676	1.0×10^{-12}	0.017269	0.026037	1.312	3.827
PD	0.3	0.006974	0.008348	0.000085	0.019132	0.026917	0.812	2.696
PD	0.5	0.007105	0.008399	0.001315	0.019271	0.027033	0.786	2.650
PD	0.7	0.007114	0.008409	0.001356	0.019290	0.027057	0.785	2.646
DP	0.1	0.004936	0.006655	0.001082	0.015522	0.023682	1.282	3.820
DP	0.3	0.008083	0.006909	0.007061	0.017716	0.024048	0.555	2.405
DP	0.5	0.008654	0.006759	0.008195	0.017893	0.023694	0.446	2.244
DP	0.7	0.008799	0.006646	0.008425	0.017799	0.023364	0.407	2.183

Usage Notes

QN-AEI is intended as a controlled benchmark dataset for studying simulated quantum noise effects in amplitude-encoded images. The released records were generated through a computational pipeline based on standard quantum

noise channels, rather than acquired from physical quantum hardware. The dataset should therefore be interpreted as a simulation-based benchmark resource, not as a device-specific measurement dataset. This design enables matched comparison across images, channels, noise levels, and repeated realizations, while avoiding confounding factors associated with hardware calibration, device topology, and time-varying noise behavior.

Software requirements. The image-level records are stored as MATLAB `.mat` files and can be loaded in MATLAB using `load`. The released records were generated using MATLAB R2025b. Regenerating the dataset from the provided script requires MATLAB with the Image Processing Toolbox, which is used for image preprocessing functions such as `imresize`, `rgb2gray`, and `im2double`. The released generation and validation script does not require the Statistics and Machine Learning Toolbox. The image-level records are saved in MATLAB `-v7.3` format and can therefore also be accessed from Python using HDF5-compatible tools such as `h5py`. Users should note that `scipy.io.loadmat` does not generally support MATLAB `-v7.3` files. Three points are worth noting for non-MATLAB readers.

First, the dataset-level manifest `qnaei_manifest_R200_200img_BSD.mat` is saved in MATLAB `-v5` format rather than `-v7.3`; it should be opened with MATLAB’s `load` or with `scipy.io.loadmat`, not with `h5py`.

Second, MATLAB stores arrays in column-major order while HDF5, and therefore `h5py`, reads them in row-major order, so every array dimension appears transposed relative to Tables 2 and 3. For example, `psi_noisy` is documented as 4096×200 but reads as a $(200, 4096)$ array under `h5py`.

Third, `psi_noisy` is stored as a complex array for the PD and DP channels but as a real-valued array for the AD channel. The amplitude-damping Kraus operators rescale amplitudes by non-negative real factors only, so no residual imaginary component survives the global-phase-alignment step of Eq. (7); the AD records are accordingly stored as real-valued arrays. For PD the stored array is formally complex, but its imaginary part is a floating-point residual of the phase-alignment step on the order of 10^{-18} rather than a physical phase; only DP carries physically meaningful imaginary components. MATLAB’s `load` handles all three cases transparently, but code using `h5py` or another non-MATLAB HDF5 library should check whether `psi_noisy` has a compound real/imaginary dtype rather than assume it is uniformly complex.

Design choices and their implications. The images in QN-AEI were resized to a fixed resolution of 64×64 pixels before amplitude encoding, ensuring that all amplitude vectors have length 4096 and correspond to 12-qubit states. Users should note that results obtained on this dataset may

not generalize directly to other resolutions or qubit counts. The dataset does not cover all possible image classes, resolutions, or application domains and is therefore best used as a controlled benchmark for methodological development and reproducible evaluation.

Before perturbation extraction, each noisy statevector was phase-aligned to the clean reference by removing the global phase, as described in the Methods section. Users who require raw unaligned complex amplitudes for phase-sensitive analyses should regenerate the records from the provided script with the `remove_global_phase` flag set to `false`. Users performing bounded-domain distribution fitting should note that the normalized perturbation vector $\mathbf{y}$ was clipped to the open interval $(0, 1)$ using threshold $\varepsilon = 10^{-12}$. This operation changes only values at or extremely close to the boundaries; the original raw perturbations are retained in `delta_raw`, allowing users to recover the unclipped values when needed, and the clipping itself ensures numerical stability in likelihood-based computations.

Reproducibility. Dataset generation uses a base random seed of 1, from which the released script derives a deterministic seed for every image-channel-noise configuration from the source-image identifier, noise channel, and noise level; any single configuration can therefore be regenerated on its own, independent of the order or number of images processed elsewhere in the run. The condition-level seed is stored in each condition record as `condition_seed`.

Source-image terms. The source images are derived from the BSDS500 training subset. Users should follow the source dataset terms for the BSDS500-derived image components. The released provenance file `source_image_provenance.csv` documents the mapping between BSDS500 source image identifiers and QN-AEI records, together with source-image metadata and applied preprocessing steps. Users should cite the original BSDS500 source when reusing or extending the dataset.

Reuse scenarios. The dataset supports several research directions, including quantum noise modeling and statistical characterization of amplitude perturbations [17, 18], evaluation of denoising and restoration approaches [14, 15], uncertainty and reliability analysis using the repeated realizations, and benchmarking of simulation-based quantum data methods. The normalized perturbation samples are directly suitable for bounded-domain distribution modeling and comparative evaluation of statistical fitting procedures [17, 18]. Comparisons with other datasets or future extensions should account for the specific preprocessing and simulation choices used to generate QN-AEI, including grayscale conversion, bicubic resizing, intensity scaling, ℓ_2 normalization, amplitude encoding, the selected noise channels and levels,

deterministic condition-level seeding, global phase alignment, and the use of $R = 200$ repeated realizations per condition.

Accompanying resources. The metadata, manifest file, validation outputs, quantitative summary statistics, and MATLAB generation and validation script are part of the dataset release and should be used together with the image-level records. The metadata and manifest files provide the mapping between source images, record files, noise channels, noise levels, and stored variables. The validation outputs document structural, completeness, dimensionality, bounded-domain, and file-readability checks. The script provides the complete workflow needed to inspect, reproduce, or extend the released records.

Data Availability

The QN-AEI dataset described in this article is deposited in *Zenodo* under the persistent identifier doi:10.5281/zenodo.22697780 (Data Citation [22]).

QN-AEI is derived from images in the training subset of the Berkeley Segmentation Dataset and Benchmark 500 (BSDS500). The original BSDS500 images are not redistributed as standalone source images in this release. Users should obtain the original source images directly from the Berkeley Computer Vision Group and comply with the BSDS500 usage terms. Users of QN-AEI should cite both the present Data Descriptor and the original BSDS500 reference.

Although the original BSDS500 image files are not redistributed as standalone image files, the released QN-AEI records contain source-image-derived arrays, including the preprocessed grayscale image I , the amplitude-encoded vector `a_clean`, noisy statevector realizations, and perturbation arrays. These derived components are distributed under CC BY-NC 4.0, preserving for this release the non-commercial restriction under which the Berkeley Computer Vision Group makes the BSDS500 source images available (“for non-commercial research and educational purposes”; see <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>).

The released QN-AEI package comprises 200 image-level MATLAB records together with the dataset manifest, source-image provenance file, validation outputs, quantitative perturbation summaries, documentation, and the MATLAB generation and validation script, following the file organization, naming conventions, and record counts documented in Data Records, above.

The MATLAB script, documentation, metadata, and validation sum-

maries developed specifically for this work are released under the same CC BY-NC 4.0 license as the image-derived records, as documented in the repository’s `LICENSE.txt`. The repository documentation distinguishes the BSDS500-derived image components from the newly generated QN-AEI records, metadata, validation outputs, and code resources.

The Zenodo record and its persistent identifier are already public. For first-round peer review, the complete deposited dataset — including all currently restricted files — is available to editors and referees via an anonymous, no-account-required reviewer link supplied separately through the manuscript submission system. The deposited files will be made openly and publicly accessible within the same Zenodo record from the second round of peer review onward, and no later than upon acceptance.

Code Availability

The MATLAB code used to generate and validate the QN-AEI dataset is released together with the data, as part of the same repository record, *Zenodo*, under the persistent identifier doi:10.5281/zenodo.22697780. The code is distributed under the same CC BY-NC 4.0 terms documented in the release’s `LICENSE.txt` and described in Data Availability, above. The code corresponds to version 1.0 and was developed and tested under MATLAB R2025b. Regenerating the dataset requires MATLAB and the Image Processing Toolbox for image preprocessing functions such as `imresize`, `rgb2gray`, and `im2double`. The released generation and validation script does not require the Statistics and Machine Learning Toolbox.

The release includes the following components:

- `qnaei_generate_dataset_BSD_R200.m` — the main generation script that reads BSDS500 training images, preprocesses them, performs amplitude encoding, simulates amplitude damping, Pauli-Z dephasing, and depolarizing noise, constructs perturbation records, saves structured image-level `.mat` files, and produces validation and summary-statistics outputs.
- Local trajectory-simulation functions implemented within the main generation script, including amplitude-damping, Pauli-Z dephasing, depolarizing-Pauli trajectory simulation, single-qubit operator application, state normalization, deterministic condition-level seed generation, and toolbox-free quantile computation.

- Validation routines for structural, completeness, dimensionality, bounded-domain, condition-metadata, file-readability, and perturbation-summary checks, also implemented as local functions within the same script.
- A README file describing the execution order, required software, input and output folder structure, record naming convention, source-image provenance file, validation outputs, the relationship between the script and the released data files, and worked examples showing how to load a record, access `cond_data`, extract noisy realizations, and inspect perturbation arrays.

The code repository documents the expected input path for the BSDS500 training images, the output directory structure, the deterministic condition-level seeding strategy, and the record naming convention

```
qnaei_bsd_record_<BSDS500-image-id>_R200.mat
```

Users who wish to regenerate the dataset should first obtain the BSDS500 training images from the Berkeley Computer Vision Group and then run the generation script with the documented input and output paths.

Author Contributions

O.A. conceived the dataset, designed the data-generation workflow, implemented the simulation and validation pipeline, generated and curated the released records, prepared the metadata, provenance files, validation outputs, documentation, and accompanying code resources, and wrote the manuscript.

Competing Interests

The author declares no competing interests.

Funding

This work received no external funding.

References

- [1] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Vancouver, BC, Canada, 2001, pp. 416–423.
- [2] D. Sierra-Sosa, M. Telahun, and A. Elmaghraby, “TensorFlow Quantum: Impacts of quantum state preparation on quantum machine learning performance,” *IEEE Access*, vol. 8, pp. 215246–215255, 2020, doi: 10.1109/ACCESS.2020.3040798.
- [3] P. Q. Le, F. Dong, and K. Hirota, “A flexible representation of quantum images for polynomial preparation, image compression, and processing operations,” *Quantum Information Processing*, vol. 10, no. 1, pp. 63–84, 2011, doi: 10.1007/s11128-010-0177-y.
- [4] Y. Zhang, K. Lu, Y. Gao, and M. Wang, “NEQR: A novel enhanced quantum representation of digital images,” *Quantum Information Processing*, vol. 12, no. 8, pp. 2833–2860, 2013, doi: 10.1007/s11128-013-0567-z.
- [5] H. Gharibyan, H. Karapetyan, T. Sedrakyan, P. Subasic, V. P. Su, R. H. Tanin, and H. Tepanyan, “Quantum image loading: Hierarchical learning and block-amplitude encoding,” arXiv preprint arXiv:2504.10592, 2025, doi: 10.48550/arXiv.2504.10592.
- [6] Z. Erkan, J. Rahebi, and A. Yelghi, “Quantum image dataset transform (QIDT) for image processing,” *Quantum Information Processing*, vol. 24, art. no. 156, 2025, doi: 10.1007/s11128-025-04754-1.
- [7] M. Lisnichenko and S. Protasov, “Quantum image representation: A review,” *Quantum Machine Intelligence*, vol. 5, art. no. 2, 2023, doi: 10.1007/s42484-022-00089-7.
- [8] M. T. West, S. M. Erfani, C. Leckie, M. Sevier, L. C. L. Hollenberg, and M. Usman, “Benchmarking adversarially robust quantum machine learning at scale,” *Physical Review Research*, vol. 5, art. no. 023186, 2023, doi: 10.1103/PhysRevResearch.5.023186.
- [9] G. Singh, H. Jin, and K. M. Merz Jr., “Benchmarking MedMNIST dataset on real quantum hardware,” *Scientific Reports*, vol. 16, art. no. 9017, 2026, doi: 10.1038/s41598-026-35605-3.

- [10] E. Perrier, A. Youssry, and C. Ferrie, “QDataSet, quantum datasets for machine learning,” *Scientific Data*, vol. 9, art. no. 582, 2022, doi: 10.1038/s41597-022-01639-1.
- [11] M. D. Wilkinson *et al.*, “The FAIR guiding principles for scientific data management and stewardship,” *Scientific Data*, vol. 3, art. no. 160018, 2016, doi: 10.1038/sdata.2016.18.
- [12] M. Barker *et al.*, “Introducing the FAIR principles for research software,” *Scientific Data*, vol. 9, art. no. 622, 2022, doi: 10.1038/s41597-022-01710-x.
- [13] O. Al-Ta’ani, A. M. Alqudah, and M. Al-Bzoor, “Implementation and analysis of quantum Fourier transform in image processing,” *Jordan Journal of Electrical Engineering*, vol. 5, no. 1, pp. 11–26, 2019.
- [14] O. Al-Ta’ani, “Hybrid quantum-classical convolutional networks for robust denoising of quantum images in noisy systems,” *International Journal of Innovative Research and Scientific Studies*, vol. 8, no. 3, pp. 1607–1618, 2025, doi: 10.53894/ijirss.v8i3.6842.
- [15] T. Kit, K. Pov, K. Kea, W.-D. Chang, H. C. Park, and Y. Han, “Enhancing a convolutional autoencoder with a quantum approximate optimization algorithm for image noise reduction,” *Machine Learning: Science and Technology*, vol. 6, no. 4, art. no. 045027, 2025, doi: 10.1088/2632-2153/ae1656.
- [16] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O’Brien, “Quantum error mitigation,” *Reviews of Modern Physics*, vol. 95, art. no. 045005, 2023, doi: 10.1103/RevModPhys.95.045005.
- [17] O. Al-Ta’ani and M. M. Gharaibeh, “A one-parameter statistical model for quantum noise-induced perturbations in amplitude-encoded images,” *IEEE Access*, vol. 14, pp. 43709–43718, 2026, doi: 10.1109/ACCESS.2026.3674019.
- [18] O. Al-Ta’ani, “Complexity-aware comparison of bounded-domain models for quantum noise effects in amplitude encoding,” *IEEE Access*, vol. 14, pp. 81460–81471, 2026, doi: 10.1109/ACCESS.2026.3697483.
- [19] J. Dalibard, Y. Castin, and K. Mølmer, “Wave-function approach to dissipative processes in quantum optics,” *Physical Review Letters*, vol. 68, no. 5, pp. 580–583, 1992, doi: 10.1103/PhysRevLett.68.580.

- [20] M. B. Plenio and P. L. Knight, “The quantum-jump approach to dissipative dynamics in quantum optics,” *Reviews of Modern Physics*, vol. 70, no. 1, pp. 101–144, 1998, doi: 10.1103/RevModPhys.70.101.
- [21] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge, U.K.: Cambridge University Press, 2010.
- [22] O. Al-Ta’ani, “QN-AEI: Quantum Noise Effects in Amplitude-Encoded Images (version 1.0),” *Zenodo*, 2026, doi: 10.5281/zenodo.22697780.