QUANTUM COMPUTING INC

Dirac-3S User Guide

September 2026, Revision A

Quantum Computing Inc
quantumcomputinginc.com
(703) 436-2161

Contents

1 What is Dirac-3S? 6
- 1.1 An Alternative Computing Approach . . . 6
- 1.2 Entropy Quantum Computing . . . 6
- 1.3 Beyond Ising Model . . . 6
- 1.4 A Specialized Hardware for Existing Computing Workflows . . . 7
- 1.5 A Scalable Hardware Platform . . . 9

2 Dirac-3 at a Glance 10

3 How Dirac-3S Computes 12

4 Quick Start/Writing an Initial Program for Dirac-3 14

5 Solve better with Dirac-3S 17
- 5.1 Summation Constraint on Continuous Solver . . . 17
- 5.2 Parameters Control . . . 18
- 5.3 Dynamic Range and Solution Quality . . . 18
- 5.4 Mean Photon Number and Shot Noise . . . 19
- 5.5 Relaxation Schedules . . . 21
- 5.6 Number of Samples . . . 21
- 5.7 Understanding Probabilistic Results . . . 21
- 5.8 Understanding Dirac-3S Timing . . . 23

6 Hardware Installation 24
- 6.1 Unpacking Dirac-3S . . . 24
- 6.2 Hardware Installation . . . 25

7 Mechanical Specifications 35

8 Related Technical References 37

9 Contacting Support 37

Copyright 2026 Quantum Computing Inc.

Document Revision History

Version	Date	Changes
Rev. A	September, 2026	N/A

Copyright 2026 Quantum Computing Inc.

Notice and Disclaimer

This User Guide, including all text, images, diagrams, and other content contained herein, is the intellectual property of Quantum Computing Inc. All rights, including copyright, are reserved. No part of this User Guide may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language without the prior written permission of Quantum Computing Inc.

Any unauthorized use or reproduction of this User Guide may violate copyright laws and international treaties. The contents of this User Guide are provided for informational purposes only and may be subject to change without notice. Quantum Computing Inc. assumes no responsibility for errors or omissions in this document or for any damages, whether direct, indirect, incidental, consequential, or otherwise, arising from the use or inability to use the information contained herein.

All trademarks, trade names, logos, and service marks mentioned in this User Guide belong to their respective owners and are hereby acknowledged.

The use of this User Guide is limited to the intended purpose of assisting users in the proper installation, operation, and maintenance of the associated hardware and software. Any other use, including but not limited to commercial exploitation, distribution, or modification of the content, is strictly prohibited.

By accessing and using this User Guide, you acknowledge and agree to respect the copyright and intellectual property rights of Quantum Computing Inc. If you have any questions or requests regarding the use of this User Guide, please contact support@quantumcomputinginc.com

Warning

⚠**Non-Compliance Notice:** This device is undergoing the process to become compliant with CE or FCC emissions certification testings. Users should take appropriate precautions and use the device at their own risk.

Future Compliance Updates: QCi is committed to continuous improvement and will release updates or revisions to enhance compliance with standards and regulations. Users are advised to regularly check for updated documents on our website and follow any recommendations provided by Quantum Computing Inc.

Contact Information: All questions or concerns regarding the compliance status or safe use of this device should be directed to the QCi customer support team.

Copyright 2026 Quantum Computing Inc.

1 What is Dirac-3S?

Dirac-3S is a specialized quantum optimization machine based on Entropy Quantum Computing (EQC) technology. It is designed to address complex, highly interconnected optimization problems by mapping mathematical objective functions into the evolution of a physical photonic system.

1.1 An Alternative Computing Approach

As optimization problems grow in size, connectivity, and complexity, finding high-quality solutions can be come increasingly computationally demanding for conventional digital computers. Alternative computing takes a different approach: rather than relying solely on sequences of digital operations, a mathematical problem is mapped onto a physical system whose natural evolution performs part of the computation. Approaches such as quantum annealing, coherent Ising machines, reservoir computing, and other physics-based computing architectures follow this analog principle [8, 3, 4]. Dirac-3S applies this concept through EQC, using the evolution and measurement of an open quantum-photonic system as a computational resource for optimization.

1.2 Entropy Quantum Computing

Unlike quantum computing approaches that seek to isolate quantum states from their environment, EQC is designed as an open quantum system, where quantum fluctuations, loss, and feedback participate in the computation. Quantum information is represented using photon-number states across optical modes, enabling Dirac-3S to operate at room temperature without cryogenic cooling. Dirac-3S implements EQC as a hybrid analog system combining quantum optics and digital electronics. Optimization problems are encoded into the photonic system, which evolves toward low-energy states representing candidate solutions. Because the computation is physical and probabilistic, Dirac-3S is an analog solver rather than an exact deterministic solver [6].

1.3 Beyond Ising Model

Many quantum and physics-inspired optimization systems are based on the Ising model, in which variables are binary and interactions are typically limited to pairwise, quadratic terms. Problems that do not naturally fit in this representation may require additional encoding, auxiliary variables, or reformulation, increasing the size and complexity of the problem presented to the hardware. Dirac-3S uses a more general polynomial objective-function representation. It supports non-binary variables, all-to-all connectivity, and higher-order interactions, allowing many problems to be represented more directly. Binary QUBO problems remain supported as a special cases, while integer, mixed discrete, quasi-continuous, and higher-order formulations can be expressed without necessarily reducing the original problem to a binary pairwise model.

Copyright 2026 Quantum Computing Inc.

Conventional Ising formulation	Dirac-3S
Binary variables	Binary, integer, mixed discrete and quasi-continuous
Pairwise interaction	Higher-order interactions
Problem may require embedding/reformulation	More direct polynomial representation
Connectivity topology-dependent	All-to-all connectivity

1.4 A Specialized Hardware for Existing Computing Workflows

Dirac-3S is not intended to replace the user's existing computing infrastructure. It is a specialized hardware designed to be integrated into existing optimization, high-performance computing, and AI workflows. Applications can use conventional CPUs and GPUs for data preparation, orchestration, and other computational tasks, while directing suitable computationally hard optimization problems to Dirac-3S.

Within an application pipeline, Dirac-3S can therefore be used to:

- Offload difficult optimization subproblems within larger classical workflows.
- Accelerate optimization-intensive AI and machine-learning workflows, including suitable optimization steps arising during model training.
- Complement CPU/GPU computing as a specialized accelerator rather than requiring an application to be redesigned entirely around quantum hardware.
- Return candidate solutions to the existing software pipeline for validation, post-processing, or downstream computation.

Copyright 2026 Quantum Computing Inc.

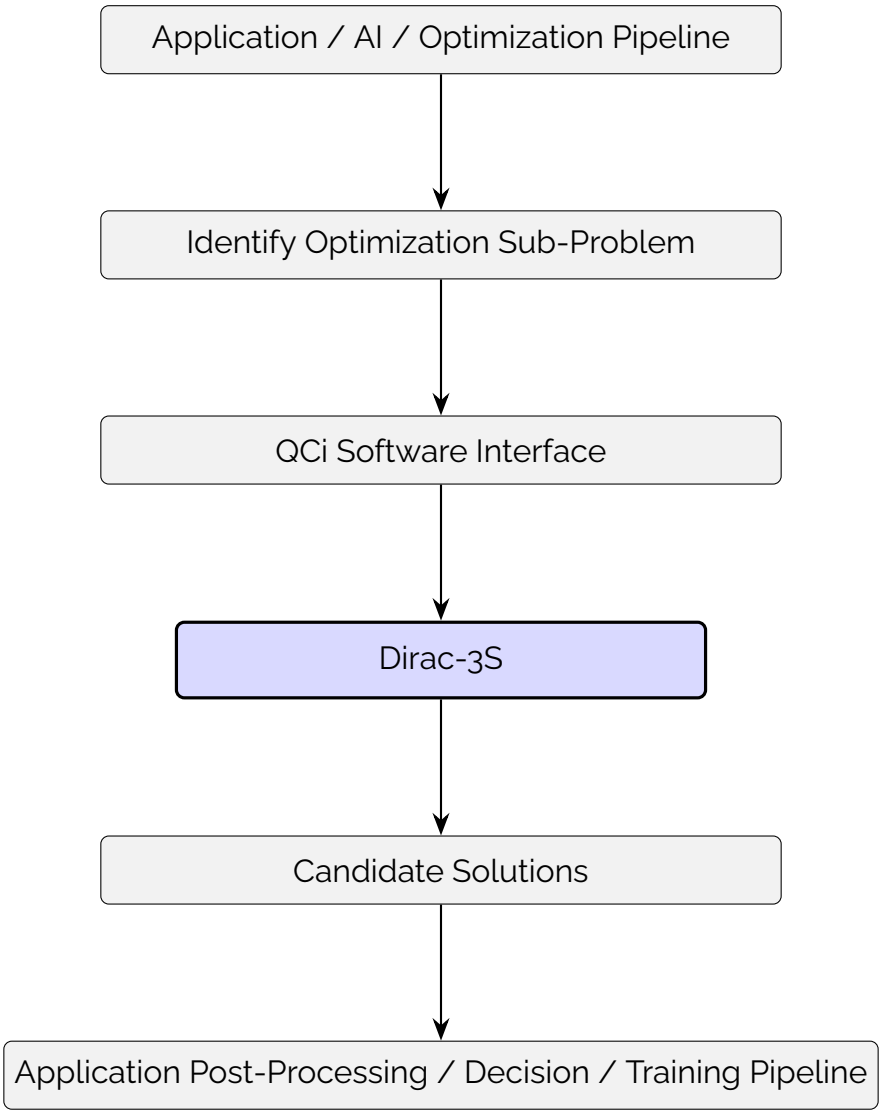

Figure 1: Using the Dirac-3S as an Optimization Accelerator within an Application Workflow.

Copyright 2026 Quantum Computing Inc.

1.5 A Scalable Hardware Platform

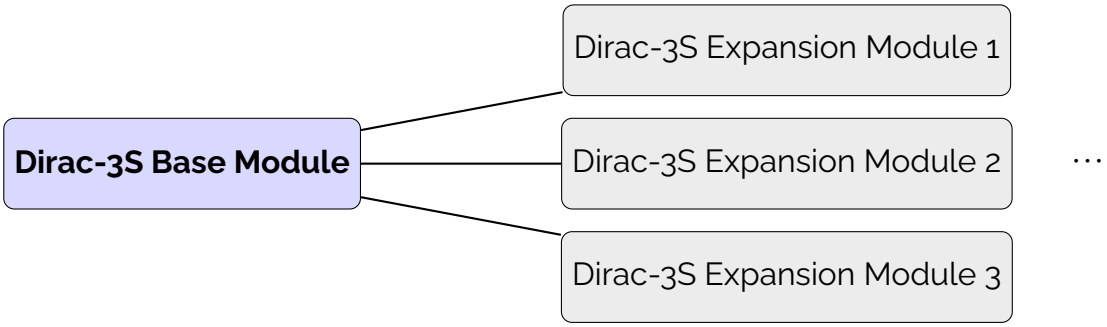

Figure 2: The Dirac-3S platform provides a scalable architecture in which a single 5U base module can be connected to one or more expansion modules to increase system capacity in problem sizes.

The Dirac-3S is designed with a modular and scalable architecture. The standard configuration consists of a compact 5U Base Module that provides full standalone operation. As application requirements grow, additional Expansion Modules may be connected to the Base Module, scaling system capacity while maintaining a consistent deployment and management model. Multiple Expansion Modules can be incorporated within a single Dirac-3S installation to support larger optimization problem sizes and faster speed.

Copyright 2026 Quantum Computing Inc.

2 Dirac-3 at a Glance

Dirac-3S is a general-purpose non-convex polynomial optimization system supporting binary, integer, mixed discrete, and quasi-continuous variables. Optimization problems are submitted to Dirac-3S as polynomial objective functions:

$$\min H = \sum_i C_i v_i + \sum_{ij} J_{ij} v_i v_j \sum_{ijk} T_{ijk} v_i v_j v_k + \sum_{ijkl} Q_{ijkl} v_i v_j v_k v_l + \sum_{ijklm} P_{ijklm} v_i v_j v_k v_l v_m \tag{1}$$

Here, v_i denotes the value of each variable. C_i represents the linear contribution (or return) of each variable, and must be a real number. J_{ij}, T_{ijk}, Q_{ijkl}, and P_{ijklm} are higher-order interaction terms, also required to be real numbers.

Dirac-3 supports direct submission for minimization problems only. To handle maximization problems, users must perform a simple transformation by multiplying the objective function by -1 before submission.

Table 1: Dirac-3S Solver Types

Solver	Variable Representation	Objective / Constraint	Best Suited For
Continuous Solver	$v_i \geq 0$, quasi-continuous	$\min_{\mathbf{v}} H(\mathbf{v}), \quad \sum_i v_i = R$	Continuous optimization with a fixed resource/sum constraint R, $R = \sum_{i=1}^{N} v_i, \quad R \in [1, 10000]$
Integer Solver	$v_i \in \{0, \ldots, 15\}$	$\min_{\mathbf{v}} H(\mathbf{v})$	Binary, integer, mixed discrete, and unconstrained discrete optimization

Copyright 2026 Quantum Computing Inc.

Table 2: Dirac-3S Hardware Characteristics and Configuration Options*

SPECIFICATION	BASE CONFIGURATION	BASE + 1 EXPANSION MODULE
System Architecture	1× Dirac-3S Base Module	1× Dirac-3S Base Module + 1× Expansion Module
Solver Characteristics		
Order of Correlation	2nd Order to 5th Order	2nd Order to 5th Order
Connectivity	All-to-all	All-to-all
Max number of variables 2nd Order	949	9980
Max number of variables 3rd Order	135	580
Max number of variables 4th Order	39	145
Max number of variables 5th Order	19	55
Physical Characteristics		
Storage Temperature	-25°C to 85°C	
Operating Temperature	20°C to 27°C	
Maximum Rate of Change	2°C per hour	
Power Consumption	100 W	< 1500 W
Network Connectivity	100/1000 Base-X Gigabit Ethernet, MM 50/125 μm Core, 850 nm, SC/PC	
Physical Dimensions	One 5U rackmount unit, 31 in depth	One 5U rackmount unit, 31 in depth + one 5U rackmount unit, 27 in depth
Software		
Software Package On-prem	eqc-direct API, Python 3.10.6	
Software Package On-cloud	qci-client API, Python ≥ 3.9 and < 3.14	
Operating System	Linux recommended	

* Additional expansion configurations above 1 module can be customized to meet larger problem sizes.

Copyright 2026 Quantum Computing Inc.

3 How Dirac-3S Computes

Dirac-3S is an analog, probabilistic hybrid optimization system that uses physical processes to search for high-quality solutions to complex optimization problems. Rather than executing a deterministic sequence of digital instructions, Dirac-3S maps the submitted objective function onto a quantum-photonic system, where quantum fluctuations, measurement, and electronic feedback collectively drive the evolution of the computation.

The objective function defines an optimization landscape in which lower-energy states correspond to better solutions. Through repeated physical evolution and measurement, Dirac-3S explores this landscape and produces candidate solutions associated with different energy values.

Dirac-3S is not an exact solver and does not guarantee discovery of the global optimum. Instead, each execution generates one or more samples from the underlying solution distribution. Repeated sampling increases the likelihood of identifying lower-energy solutions, with the lowest-energy observed sample representing the best solution found during that computation.

Copyright 2026 Quantum Computing Inc.

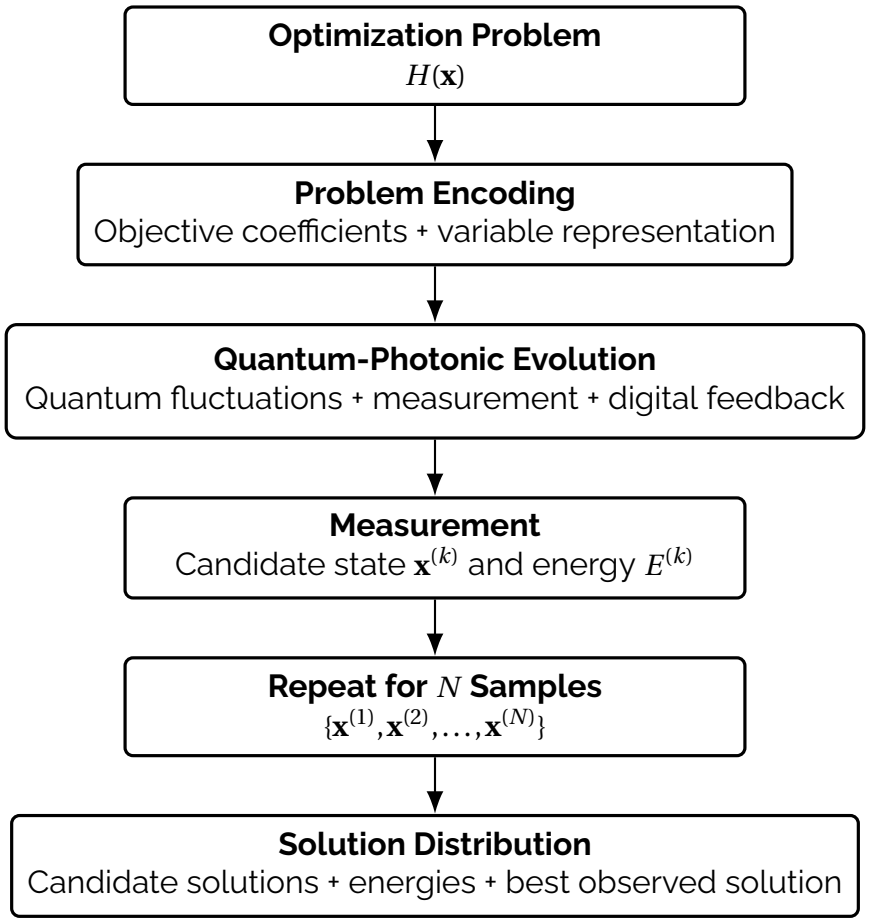

Figure 3: Conceptual computation flow of Dirac-3S. A submitted optimization problem is encoded into the quantum-photonic system, which evolves under measurement and feedback to produce candidate solutions. Repeated sampling generates a distribution of observed solutions and energies.

Copyright 2026 Quantum Computing Inc.

4 Quick Start/Writing an Initial Program for Dirac-3

Here we share a basic problem submission example for Dirac-3. All interactions with the device are resolved using eqc-direct a Python package which utilizes gRPC messaging to interface with the device. The package provides utility functions such as acquiring a lock on the device, system monitoring, testing the health of the device, submitting a problem, and cancelling a currently processing problem submission. In order to illustrate a standard problem submission a simple polynomial problem will be utilized:

$$E(x_1, x_2, x_3) = -10x_1 - 10x_3 - 3.7x_2^2 - 5x_1x_3 \ , \tag{2}$$

The first step is to extract the polynomial coefficients and format polynomial variable indices for each term in the equation. The polynomial coefficients will be represented as a list as follows:

```
poly_coefs = [-10, -10, -3.7, -5]
```

The polynomial indices for the coefficients will be represented as follows in the same order as represented in the original equation:

```
poly_indices = [[0,1], [0,3], [2,2], [1,3]]
```

Note, that each individual polynomial index set has non-decreasing values moving from left to right such that for each set of indices:

$$[k_1, k_2, k_3, \ldots, k_{n-1}, k_n] \tag{3}$$

the following condition must hold:

$$k_1 \leq k_2 \leq k_3 \leq \cdots \leq k_{n-1} \leq k_n \tag{4}$$

This property guarantees the uniqueness of the submitted polynomial index set. Also, this sparse representation minimizes the amount of data needed to be transferred to the device to obtain a solution.

After formatting the problem for submission, the client can be used to submit the problem after acquiring an exclusive execution lock on the device. The inputs to the function used to submit problems, *solve_sum_constrained*, are as follows:

- *lock_id* - a UUID string with exclusive lock for device execution
- *poly_coefficients* - the coefficient values for polynomial to be minimized. Numbers, including integers, should be floats with 32-bit (or less) precision, otherwise precision is lost during conversion to 32-bit.
- *poly_indices* - list of lists containing the indices for coefficient values for polynomial to be minimized.

Copyright 2026 Quantum Computing Inc.

- *num_variables* - optional input to specify number of variables for polynomial. Must be greater than or equal to maximum index value in *poly_indices*.
- *relaxation_schedule* - four different schedules represented in integer parameter. Higher values reduce the variation in the analog spin values and therefore, are more probable to lead to improved objective function energy for input problem. Accepts range of values in set {1, 2, 3, 4}.
- *sum_constraint* - a normalization constraint that is applied to the problem space that is used to calculate energy. This parameter will be rounded if exceeds float32 precision (e.g. 7-decimal places). Value must be between 1 and 10000.

For additional information see source documentation for function here.

```
#!/usr/bin/env python

from eqc_direct.client import EqcClient

# MUST FILL IN THE VALUES IN THIS FROM YOUR NETWORKING SETUP FOR THE DEVICE
eqc_client = EqcClient(ip_address="localhost", port="50051")
poly_coefs = [-10, -10, -3.7, -5]
poly_indices = [[0,1], [0,3], [2,2], [1,3]]
# use lock to prevent other users from taking exclusive access to the device
lock_id, start_ts, end_ts=eqc_client.wait_for_lock()
try:
    result_dict = eqc_client.solve_sum_constrained(
        lock_id=lock_id,
        poly_indices = poly_indices,
        poly_coefficients = poly_coefs,
        relaxation_schedule = 2,
        sum_constraint = 100)
finally:
    # release lock when finished using the device
    lock_release_out = eqc_client.release_lock(lock_id=lock_id)
```

Each response as obtained in *result_dict* above will include the following fields:

- *err_code* - 0 if problem was solved without any issues, otherwise an integer greater than 0 representing the error that occurred
- *err_desc* - a short description of the error code for the submission
- *preprocessing_time* - time in seconds to validate data and re-format input data for running on the device.
- *runtime* - the time in seconds which the device required to solve the problem
- *postprocessing_time* - runtime for auxiliary computations that occur besides sampling during each sample routine from Dirac hardware including intermediate energy calculations, objective function adjustments, and distillation of solutions.

Copyright 2026 Quantum Computing Inc.

- *energy* - list of energies for best solution found (float32 precision) for each sample from Dirac hardware after error correction
- *solution* - a list of vectors representing the lowest energy solution (float32 precision) for each sample from Dirac hardware after error correction
- *num_variables* - specifies the number of variables in the polynomial in solved polynomial.
- *num_samples* - the number of samples (independent solutions) to generated from the device
- *calibration_time* - calibration time is unrelated to execution of the individual sampling for the optimization. This time is from system level interruptions from calibrations that happen at regular intervals to maintain system performance
- *start_job_ts* - nanosecond timestamp marking start of submission to device
- *end_job_ts* - nanosecond timestamp marking time at which results were acquired from device and returned to the user
- *raw_energy* - list of energies for best solution found (float32 precision) for each sample from Dirac hardware before error correction
- *raw_solution* - a list of vectors representing the lowest energy solution (float32 precision) for each sample from Dirac hardware before error correction

An example response for the problem above would appear as follows:

```
{'calibration_time': 0.0,
 'end_job_ts': 1789487196956684063,
 'energy': [-60.0],
 'err_code': 0,
 'err_desc': 'Success',
 'num_samples': 1,
 'num_variables': 3,
 'postprocessing_time': [1.4899999314366141e-06],
 'preprocessing_time': 0.004257313907146454,
 'raw_energy': [-60.0],
 'raw_solution': [[2.0005233, 0.0, 1.9994767]],
 'runtime': [0.8978748321533203],
 'solution': [[2.0, 0.0, 2.0]],
 'start_job_ts': 1789487195931085785}
```

To run a problem using the integer solver see overview here, function spec for input function *solve_integer* here, and description of function output here.

5 Solve better with Dirac-3S

5.1 Summation Constraint on Continuous Solver

*Using amplitude encoding on each photon time-bin, Dirac-3 functions as a quasi-continuous optimization solver, subject to the constraint of a fixed resource R:

$$R = \sum_{i=1}^{N} v_i, \quad R \in [1, 10000]$$

This means the resulting state vector v_i is continuous, with resolution determined by the system. Refer to section 5.3 for more details. Below is an example. We evaluated the following quadratic cost function using Dirac-3:

$$E(x) = 5v_1^2 + 4v_2^2 + 3v_3^2 + 2v_4^2 + v_5^2,$$

with the sum constraint $R = \sum_{i=1}^{5} v_i = 100$; a typical solution vector from Dirac-3 is

$$\vec{v} = [8.53924, 10.9957, 14.7423, 21.9537, 43.7687],$$

and corresponding objective value = 4379.86.

When the same Hamiltonian is optimized with a larger constraint $R = 1000$, the solution remains functionally similar, but the precision is reduced due to the fixed number of resolution levels:

$$\vec{v} = [87.5832, 109.487, 145.987, 218.989, 437.953],$$

with objective value = 437956.

This behavior reflects a fundamental tradeoff: increasing the solution space without increasing resolution granularity reduces effective precision.

**Using time-bin location encoding, Dirac-3 can also act as an unconstrained integer optimization solver. In this mode, the resulting state vector v_i consists of integers:

$$v_i \in \mathbb{N}, \quad 0 \leq v_i \leq 16$$

Users must specify the number of discrete states (levels) for each variable. For problems requiring more than 16 states, the continuous solver is recommended.

This flexible encoding allows Dirac-3 to support Quadratic Unconstrained Binary Optimization (QUBO) by setting the number of levels for each variable to 2. Furthermore, Dirac-3 supports mixed encoding, enabling some variables to have binary values (0 or 1) and others to have higher integer levels than 2.

Copyright 2026 Quantum Computing Inc.

5.2 Parameters Control

Dirac-3 exposes a set of control parameters that allow users to customize the optimization process for different problem classes and performance requirements. The available parameters, valid ranges, and default settings are summarized in Table 3.. Subsequent sections describe the effect of each parameter on solver behavior and solution quality.

Table 3: Dirac-3 control parameters, valid ranges, and default settings.

Parameter	Description	Range	Default
Dynamic Range	Controls the effective signal dynamic range used during computation.	≤ 23 dB	System-defined
Quantum Fluctuation Coefficient	Controls the magnitude of fluctuations used to explore the solution space.	1–100	1 (Continuous), 15 (Integer)
Mean Photon Number	Sets the average optical energy level during optimization.	6.67×10^{-5} to 6.67×10^{-3}	3.33×10^{-4}
Relaxation Schedule	Defines the relaxation profile used throughout the optimization process.	1, 2, 3, 4	1
Number of Samples	Number of independent solution samples generated per solver execution.	1–100	1

5.3 Dynamic Range and Solution Quality

Dynamic range of C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm} are constrained to finite resolutions from optical modulator, digital-to-analog converter (DAC), input/output signal conditioning, and variation of temperature as well as dark count from single photon detector.

The analog nature of Dirac-3 causes the system's effective dynamic range to vary due to temperature fluctuations and device-level imperfections when measuring quantum states. While Dirac-3 can accept any signal within the digital system's readout range (up to $\approx 70dB$), it struggles to distinguish very small coupling terms due to its effective analog resolution limit, which is approximately 200:1, or 23 dB.

This limitation can be mathematically expressed as:

$$\frac{max(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})}{min(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})} \leq 200$$

Furthermore, in order to resolve two distinct coupling terms, their difference must exceed the minimum resolvable level (i.e., the maximum term divided by the dynamic range). This constraint is written as:

$$| x - y | \geq \frac{max(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})}{200} \tag{5}$$

$$x, y \in \{C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm}\}, \quad i \neq j \neq k \neq l \neq m \tag{6}$$

Copyright 2026 Quantum Computing Inc.

5.4 Mean Photon Number and Shot Noise

Coherent states quantum state of light closely resemble classical oscillation of light field, except from superimposed with quantum noise, specifically shot noise. Dirac-3 quantum optimization machine uses single photon count of each time-bin as a feedback signal. Therefore, the system experience Poisson noise taken advantage as a source of "entropy" or "fluctuation", constantly being boosted into each feedback loop. This parameter is defined as "quantum fluctuation coefficient" (QFC) that equals to $\frac{1}{\sqrt{N}}$, following the standard deviation of Poisson distribution where N the number of photon collected in each time-bin. Therefore, throughout the measurement and feedback process, the injection of quantum fluctuations enables a search in a near-infinite search space, which assists in bypassing the trapping of the quantum state into local minima. Users can specify QFC by integer from 1 to 100 where higher fluctuation corresponds to lower integer value.

While our entropy computing approach implementation is designed for photon number states or Fock states, it is practically difficult with the current state-of-the-art in deterministic photon number generation [7, 5]. In current hybrid implementation of Dirac-3, we use a common approximation of Fock states which is coherent states with ultra-low mean photon number μ[2] (MPN). Dirac-3 is maintained such that about 1% or less of pulses has more than 1 photon. Users have option to modify MPN to search for best parameter for their problems between the range from 6.67×10^{-5} to 6.67×10^{-3}. Following is an study to show how MPN affects solution quality.

We evaluate a non-convex quadratic optimization problem (QPLIB_0018) with 50 continuous variables over a fully connected weighted graph, selected from QPLIB, a library of quadratic programming instances [1]. The cost function of this problem is of the form $f(\mathbf{x}) = C^T\mathbf{x} + \mathbf{x}^T J\mathbf{x}$, where $\mathbf{x} = (x_1, x_2, \cdots, x_{50})^T$ and $x_i \geq 0$.

Copyright 2026 Quantum Computing Inc.

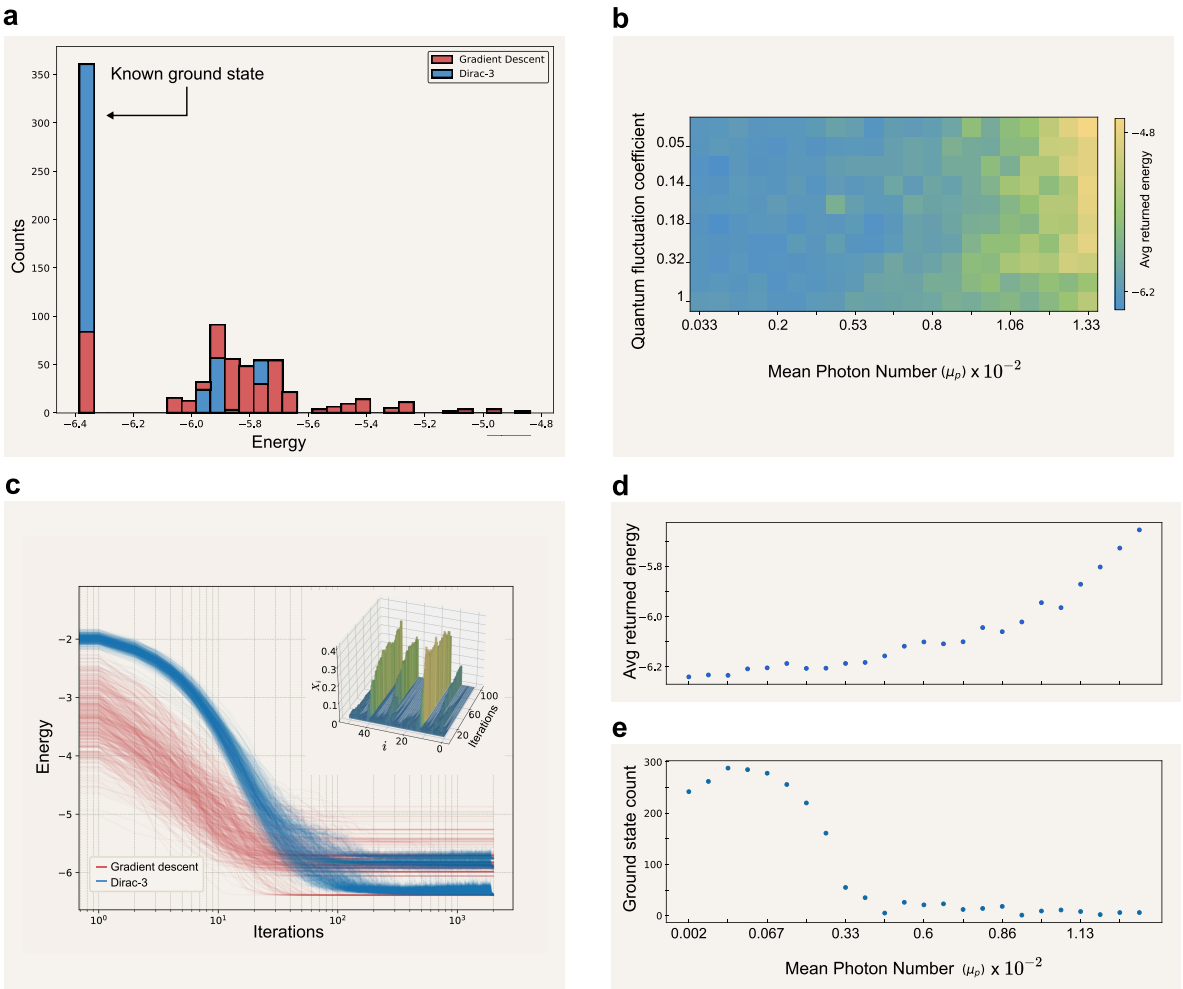

Figure 4: **Solving a non-convex continuous optimization problem.** A non-convex quadratic optimization problem with 50 variables is considered (QPLIB_0018). (a) Energy distribution over 500 runs of Dirac-3 (blue) and gradient descent algorithm (red). (c) Energy evolution versus the number of iterations on Dirac-3 (blue) and gradient descent (red). The inset shows the evolution of the solution versus iterations. (b) Investigate the relationship between the mean photon number (μ), quantum fluctuation coefficient,and key performance metrics - average returned energy. The transitioning from left to right, represents an increase in mean photon number (μ), showing the probability that Dirac-3 is operating in single-photon regime. Quantum fluctuation or shot noise is used as $\frac{1}{\sqrt{N}}$ where N is the number of photon count accumulated in each time-bin. From bottom to top of the vertical axis, this coefficient is gradually decreased. (d), (e) Average returned energy and number of ground states found after 500 runs are collected when mean photon number is decreased further up to 0.002 %, close to the level of dark count of single photon detector.

Copyright 2026 Quantum Computing Inc.

5.5 Relaxation Schedules

Dirac-3 offers four relaxation schedules, $1, 2, 3$ and 4, each determining the time allocated for the system to evolve and converge towards the ground state. Evolution time for relaxation schedule 1 is the shortest and it increases to the longest for schedule 4.

5.6 Number of Samples

Due to stochastic nature (See 5.7 for more details) of the solver, it is often useful to run the experiment multiple times and find the best solution among those runs. This can be done by setting "num_samples" to an integer greater than 1. User devices are not subject to the sample maximum of 100 that exists for cloud devices.

5.7 Understanding Probabilistic Results

An analog computing system maps a mathematical problem into physical phenomena and therefore it is a natural fit optimization purposes. However, that comes with a device dependency, due to both inevitable errors of the physical hardware and the inherent non-deterministic nature of the measurement outcomes. Therefore, it is not unusual that even for a simple problem that can be solved in digital computer, Dirac-3 will have some probability of returning a sub-optimal answer. For example, consider the following objective.

$$E(x_1, x_2, x_3) = -10x_1 - 10x_3 - 3.7x_2^2 - 5x_1x_3 \ , \tag{7}$$

subject to a sum constraint $x_1 + x_2 + x_3 = 4$. This objective has a local minimum $E(0, 4, 0) = -59$, and global minimum $E(2, 0, 2) = -60$. Dirac-3 solver has the following energy distribution in Figure 5 after 100 trials.

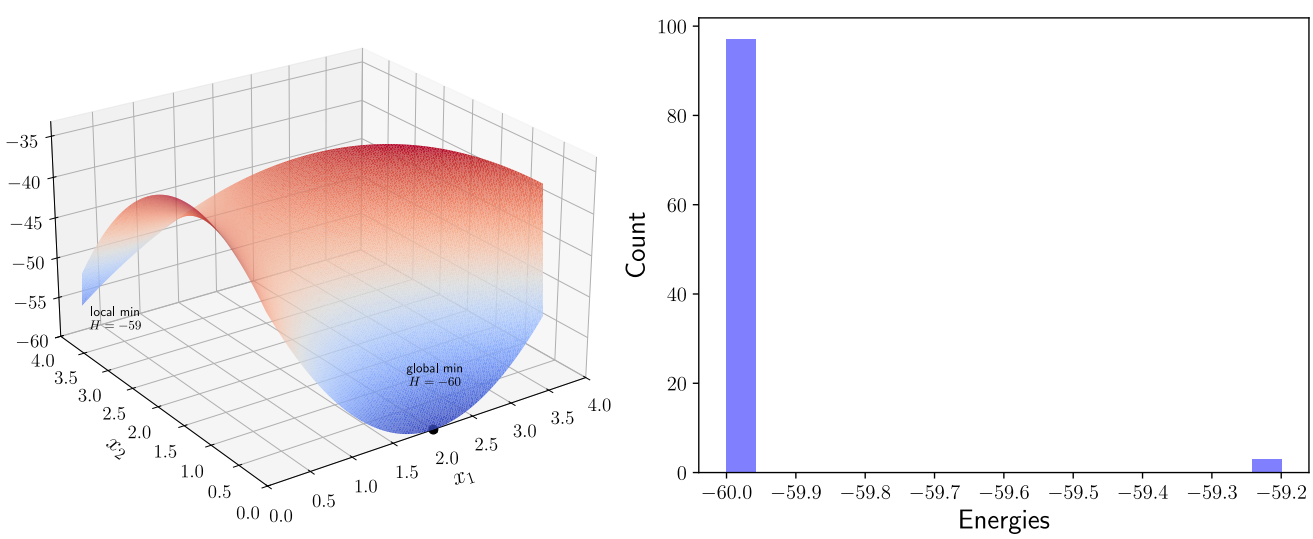

Figure 5: Left: Landscape of objective from Eqn. 7. Right:Energy distribution results from optimization using Dirac-3. Dirac-3 returns optimal solution 97% and sub-optimal 3% after 100 trials.

Copyright 2026 Quantum Computing Inc.

In Figure 6, a non-convex problem from the Library of Quadratic Programming Instances QPLIB_0018 with a known optimal energy -6.386 was solved using gradient descent. Due to the non-convex nature of the problem, it is expected that the gradient descent has some probability of getting to global optimum and some probability of falling into a local optimum, depending on the starting point of the search.

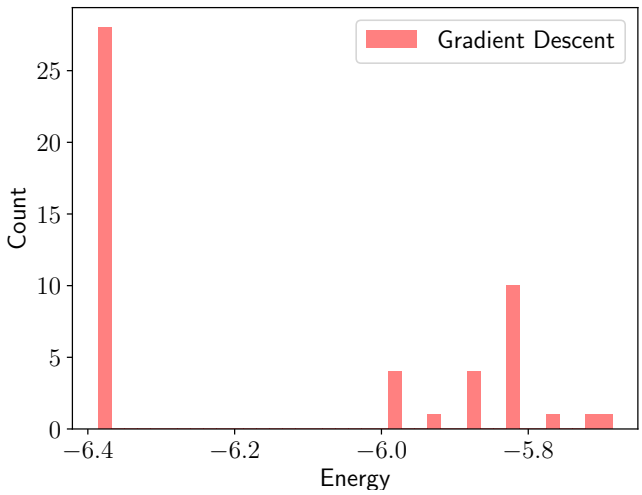

Figure 6: Energy distribution from gradient descent with 50 random initial starts. This shows that gradient descent has some probability of getting to global optimum and some probability of falling into a local optimum, depending on the starting point of the search

Taking advantage of quantum fluctuation, Dirac-3 is expected to be able to jump out of local minimum valleys with high probability. The same problem was run on Dirac-3 50 times with schedule 1, 2, 3, and 4. The energy distribution for each schedule is shown in Figure 7. We see that more states were able to escape local minima as we increased relaxation schedule.

Copyright 2026 Quantum Computing Inc.

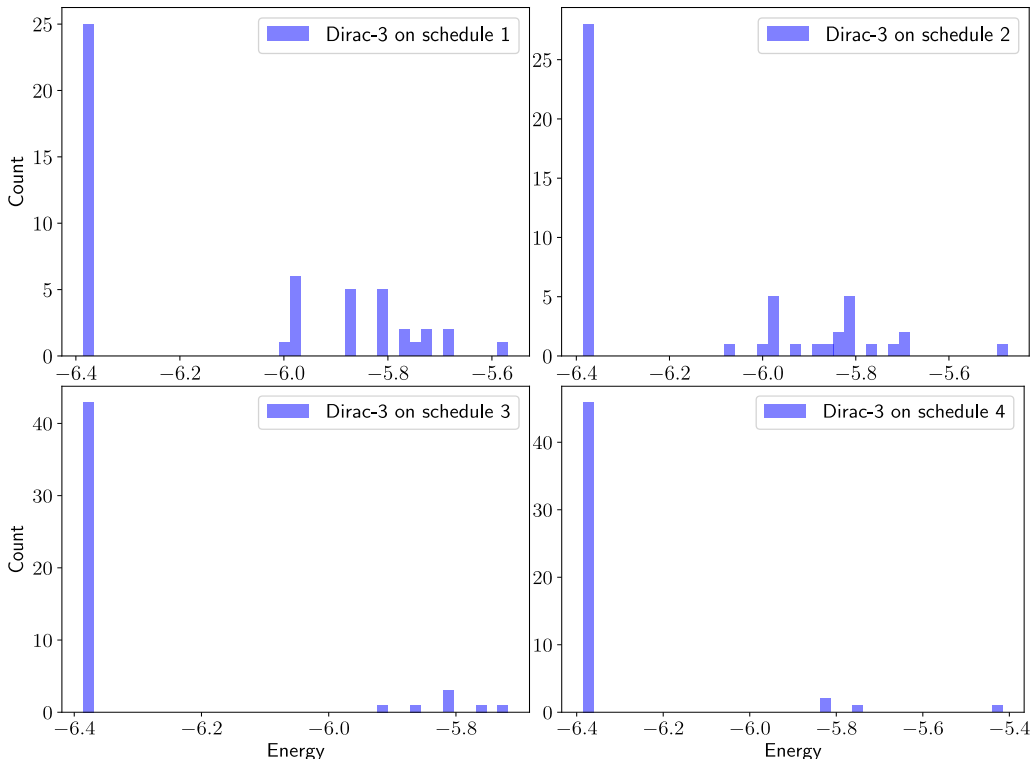

Figure 7: A typical distribution of energy over 50 runs. Dirac-3 samples more optimal states as we increase the relaxation schedule.

A comparison of the average computing time over 100 samples on a basic 50 variable sum constraint problem without pre and post processing for schedules 1 through 4 can be found in table 4.

Schedule	Average time per sample (s)
1	0.236
2	0.420
3	1.093
4	1.278

Table 4: Relaxation Schedule vs. Average execution time per sample in a simple sum constraint problem

5.8 Understanding Dirac-3S Timing

Dirac-3 leverages digital electronics through the utilization of a field-programmable gate array (FPGA) to process quantum wavefunction measurements information. Matrix multiplication is used to emulate the global interaction of each "qudit," with results feeding back into the quantum system and undergoing multiple iterations sequentially. Consequently, the overall computing time is the cumulative sum of each iteration.

Copyright 2026 Quantum Computing Inc.

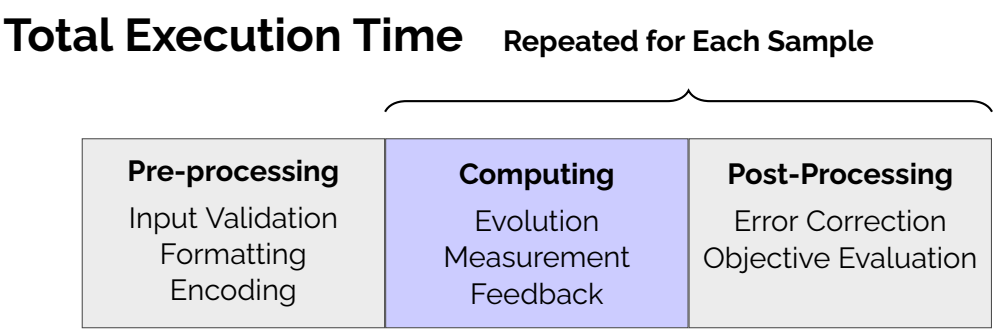

Figure 8: Conceptual execution-time breakdown for a Dirac-3S computation when it is connected directly to a single host server/Computer. Pre-processing occurs once at the start of the job. Computing and post-processing are repeated for each requested sample.

6 Hardware Installation

6.1 Unpacking Dirac-3S

Follow these steps:

1. Unpack and remove the device and the accessory kit from the shipping box.
2. Return the packing material to the shipping container, and save it for future use.
3. Verify that you have received the items shown below. If any item is missing or damaged, contact your QCi Support representative or reseller for instructions.

Copyright 2026 Quantum Computing Inc.

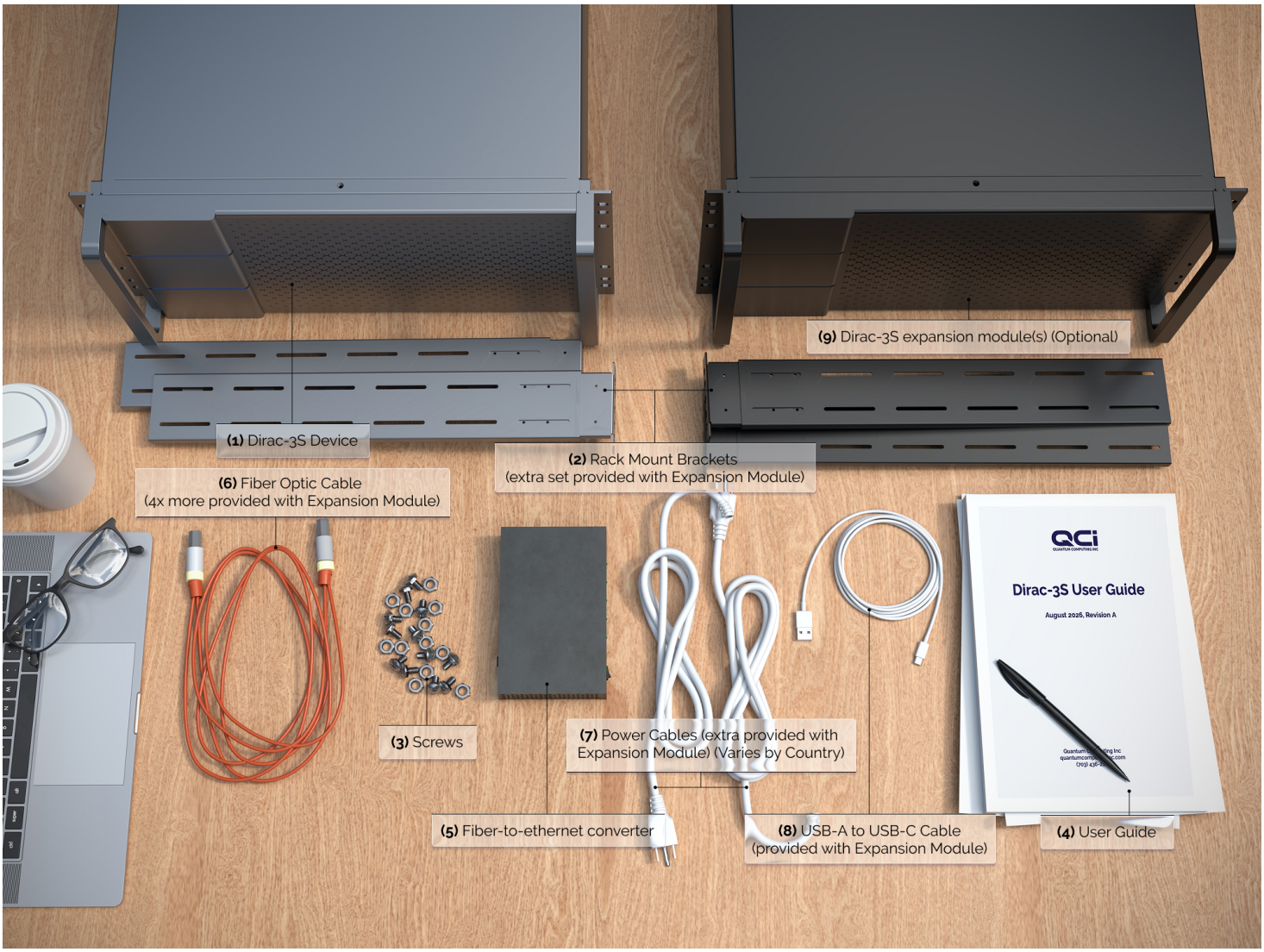

Figure 9: Items included with the Dirac-3S system shipment.

6.2 Hardware Installation

6.2.1 Equipment and Environment Requirements

When setting up your device it is necessary to choose an appropriate location which can meet the following requirements for operation:

- At least 5U of rack mounting space for the base configuration and extra 5U for each additional Expansion Module.
- Unrestricted airflow around the server and its vents.
- Temperature must not fall below 20°C or exceed 27°C around the server. Be mindful of

heating and A/C vents nearby that may blow directly on the device.

- Humidity must not exceed 85%.
- All cables must be kept away from sources of electrical noise, such as radios, power lines, and fluorescent lighting fixtures.
- The cable length from a switch to an attached device cannot exceed 328 feet (100 meters) as Ethernet Specified.

6.2.2 Safety Precautions

⚠When installing Dirac-3S, at least TWO people are required to safely and easily lift and move the unit. Installation with one person is NOT recommended!
⚠Dirac-3S utilizes class 3B/3R Lasers which are dangerous for human eyes! DO NOT open the system!
⚠Do not operate Dirac-3S in an area that exceeds the maximum recommended ambient temperature of 27˚C to prevent overheating of the server. Allow at least 3 inches (7.6 cm) of clearance around the ventilation openings to prevent airflow restriction.

Installation of the equipment must comply with local and national electrical codes.

6.2.3 Server Rack Installation

This section describes the procedure when mounting a Dirac-3S system into a server rack. While not required for normal operation, it is recommended to install the system in a temperature controlled, well ventilated area such as a server room in order to meet operational environment requirements. Dirac-3S comes with one 5U system for base configuration and extra 5U for each additional Expansion Module), following process explains how to mount base unit, repeat same process with expansion module

Prepare the Device: Ensure all units are powered off and disconnected from power sources. Remove any packaging materials or protective covers from the device.

Attach Server Rack Rails: Attach and secure server rack rails to both sides of the vertical post of the rack. See figure 10 and figure 11 (Note: Rails are adjustable between 30 and 35 inches in length; the distance between the front and back vertical rails on the rack must be in this range.)

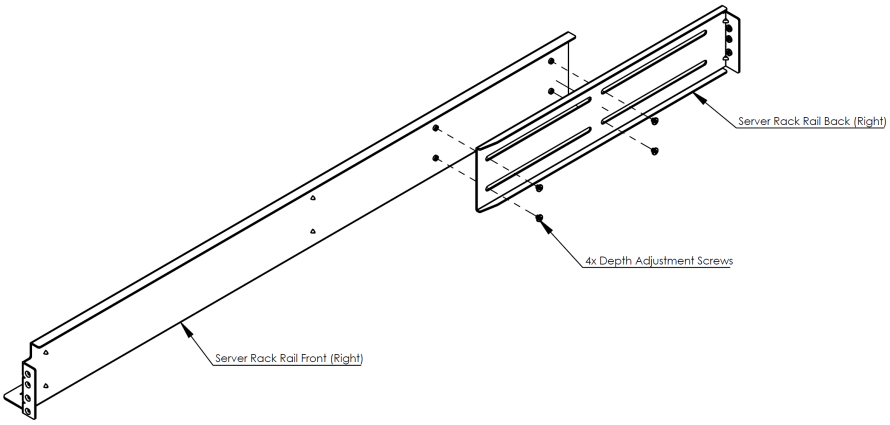

Figure 10: Server rack rails front and back

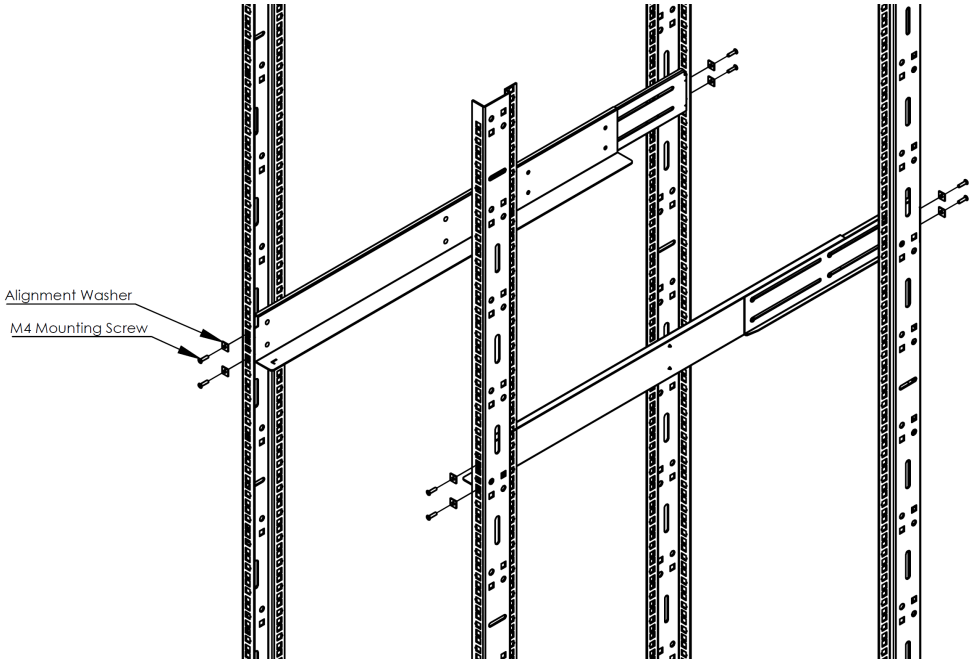

Figure 11: Mounting rails

Position and Secure the Device: Carefully carry the device and gently slide it into the desired 5U slot in the rack. Ensure the position of Dirac-3S device parallel to the floor while moving. Once the device is positioned correctly, use screws to secure the mounting rails to the rail support. See figure 12 and 13. Make sure the device is level and aligned properly within the rack.

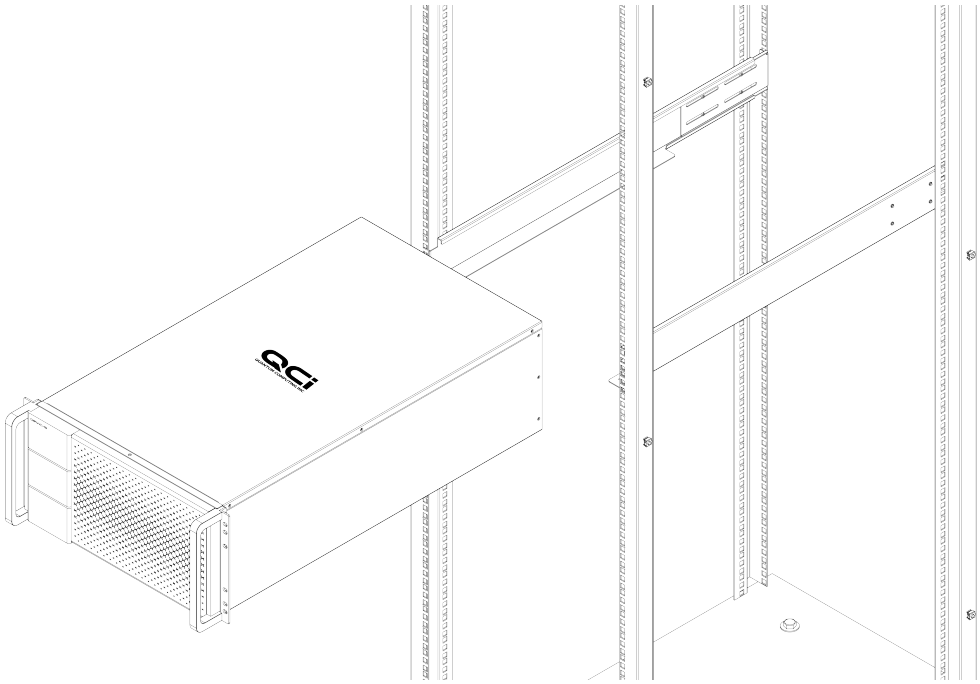

Figure 12: Positioning the Dirac 3S Base Unit

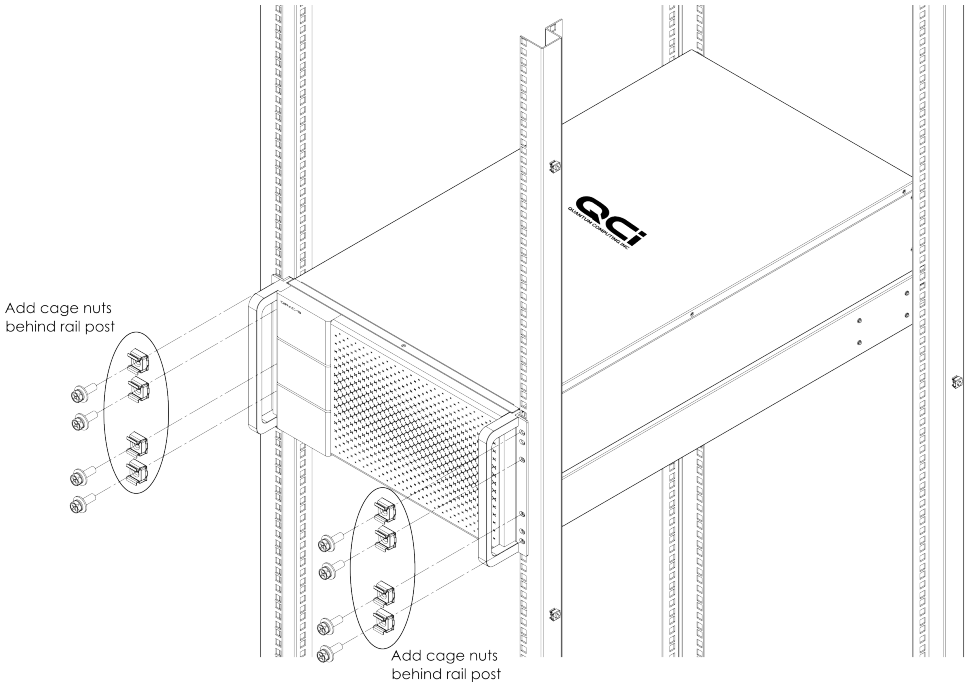

Figure 13: Securing the Dirac 3S Base Unit

Repeat Process: Repeat the same process for the Expansion Module. Once completed, setup should look like that provided in figure 14

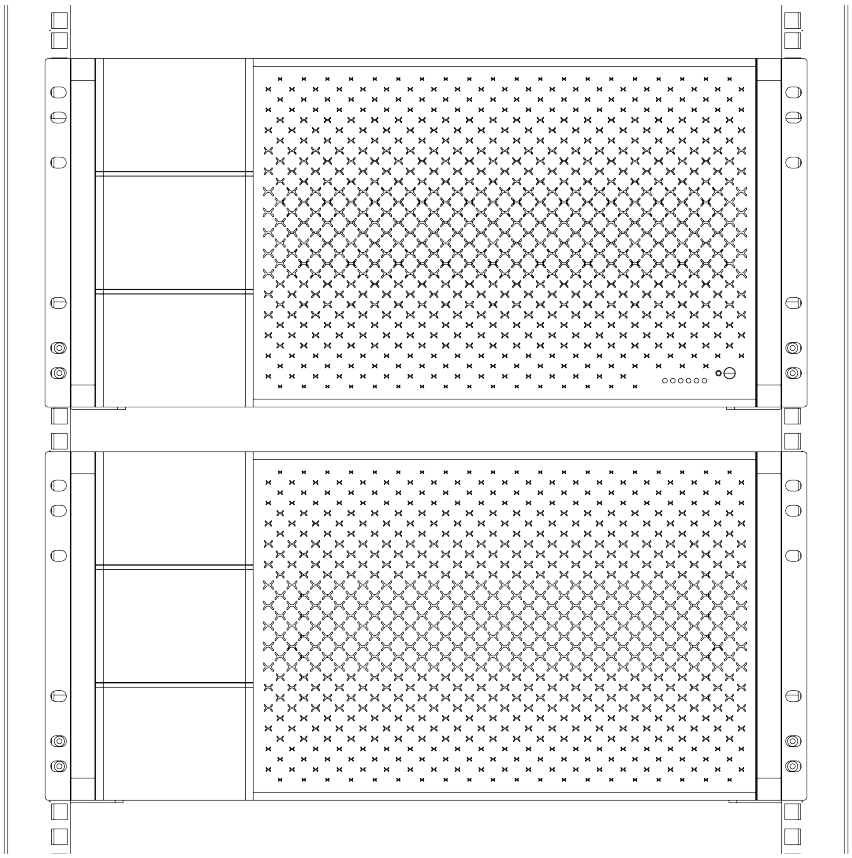

Figure 14: Dirac-3S in Rack

Cable Management: Organize power cable and Ethernet cable connections coming from both 5U devices to the host server in the rack or outside the rack. Use cable management tools such as cable ties or cable management panels to keep cables neat and organized.

6.2.4 Powering on Dirac-3

Dirac-3S comes with one base module and additional one or more Expansion Module(s). The Base Module connects to the user's local area network (LAN) via one 1000M LC fiber Ethernet interface. A 1G 850nm 500m range SFP is the recommended way to interface with the system. The digital system is self contained and is not connected to the network. The Base Module and Expansion Module systems are linked via four 10G SFP+ interconnects, labeled Port1 to Port4. These must be connected together for proper operation.

To maximize operational compatibility, both systems accept universal single-phase input power. Since data center rack power is commonly delivered at 220VAC/60Hz in the United States, this is recommended for both systems. Before plugging the device into a wall outlet or a host device, check to make sure that the power switch is set to "OFF". Once verified, plug the power cord into the unit. Next, plug the power cord into the wall outlet. Power on the device by flipping the power switch. Notice that the LED bar in the front of the unit should be glowing blue-violet to indicate

Copyright 2026 Quantum Computing Inc.

power is on and the device is waiting for communication.

The Expansion Module will take longer to power up. Once plugged into a power source, the LED indicators on the redundant power supplies will turn on as well as the indicator LEDs on the front panel. Once all of the LEDs are off on the front panel, the system is ready to be powered on. Press the power button, you will see the indicator LEDs change and hear the system cooling fans kick on. After this, the Expansion Module will take about 5 minutes to boot up before its ready for use.

A diagram of the intended setup is provided in figure 17:

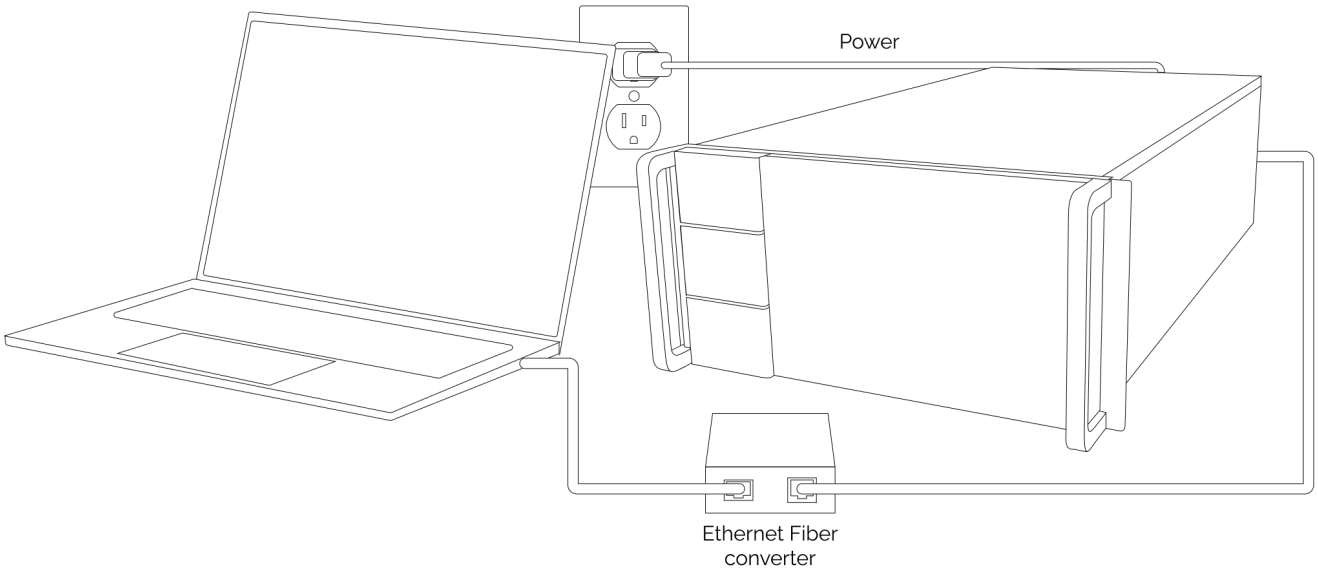

Figure 15: Example of Dirac-3S setup in the base configuration where host computer is connected directly to the device over physical Fiber/Ethernet connection

Copyright 2026 Quantum Computing Inc.

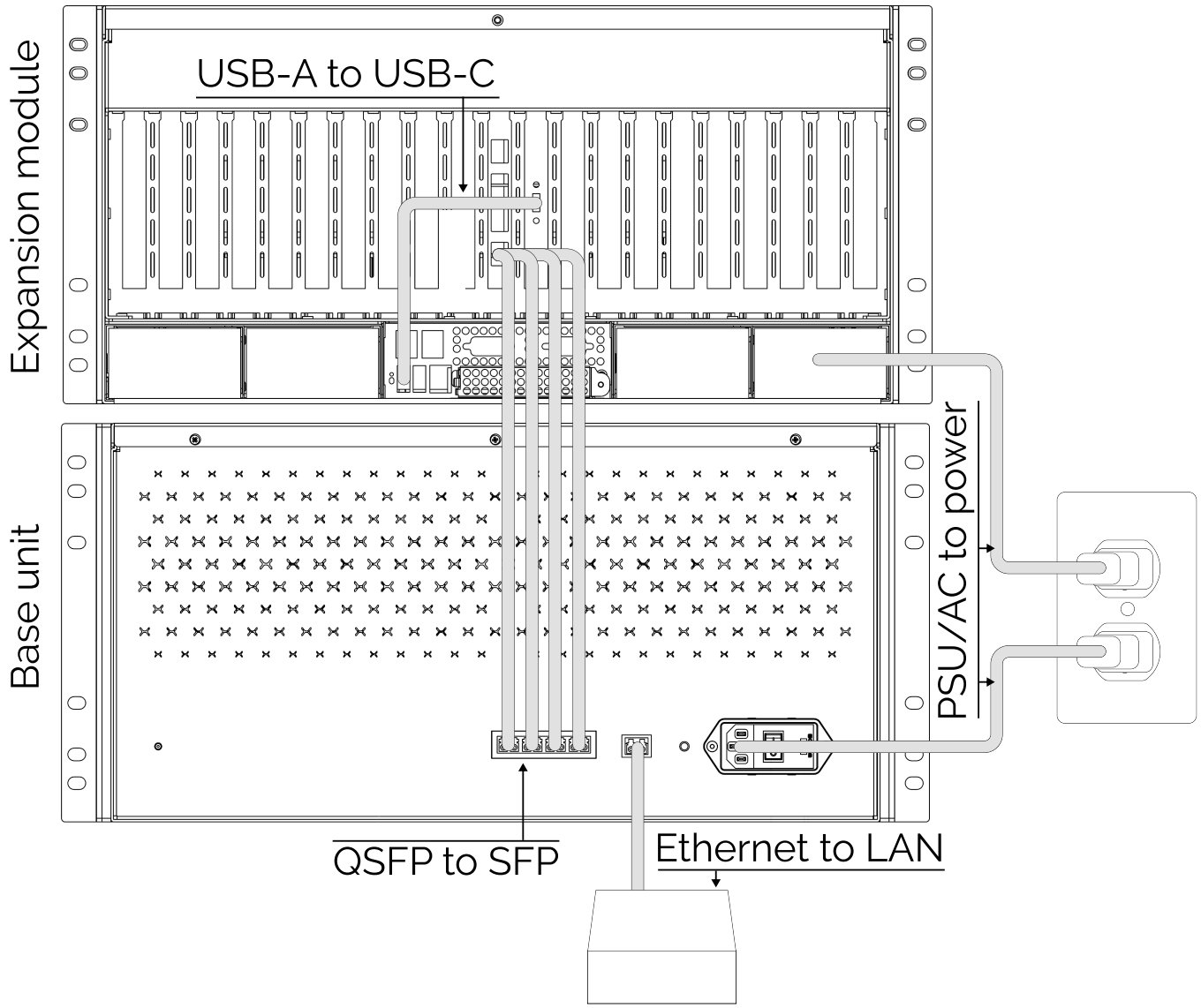

Figure 16: Example of Dirac-3S setup with 1 Expansion Module where host computer is connected directly to the device over physical Fiber/Ethernet connection

Copyright 2026 Quantum Computing Inc.

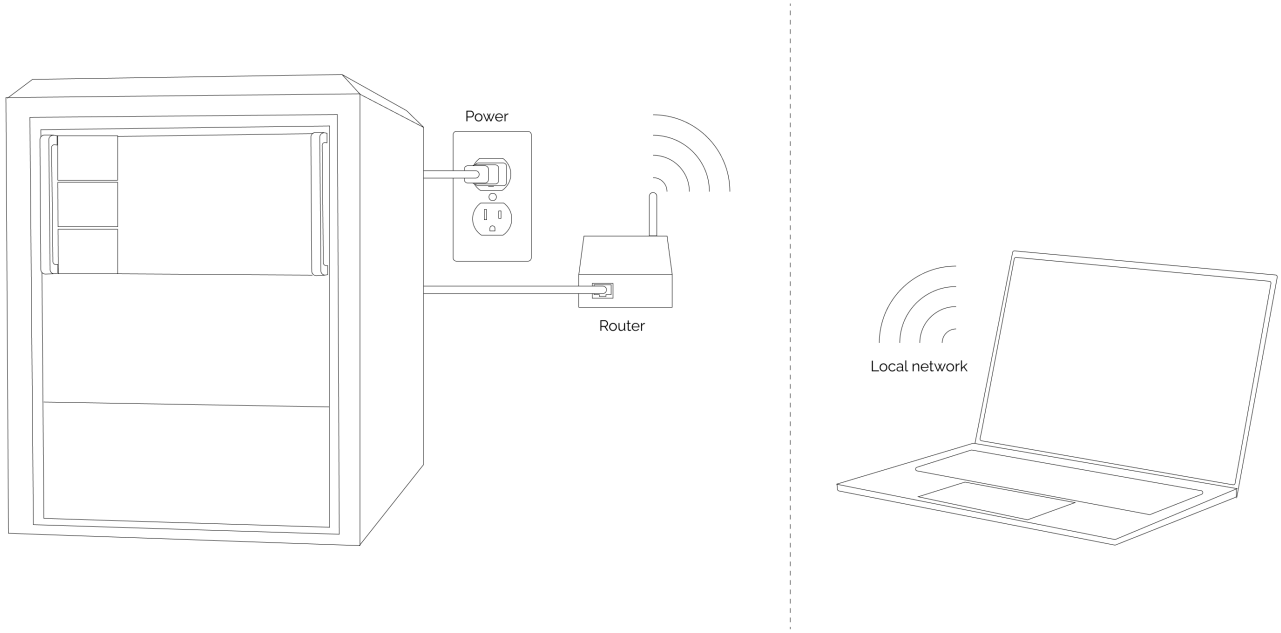

Figure 17: Example of Dirac-3S setup where the system is positioned in a standard data center rack (1 Base Module only), connected to a router. Users can access the device within local network.

Figure 21 illustrates the input/output connections for Dirac-3S.

On the Base Module, the back panel includes an AC power plug, a power switch, and five LC optical ports. Ports 1 through 4 are for linking the analog and digital systems, and port 5 is for connecting to the LAN.

On the Expansion Module, the back panel includes four redundant AC power plugs, four LC optical ports, one USB-C connector, and two USB-A connectors. Optical Ports 1 through 4 are for linking the analog and digital systems, and the USB-C and USB-A connectors are linked together.

Two LEDs located on the front panel of the analog system (left) indicate device status. See table 5 for details.

Copyright 2026 Quantum Computing Inc.

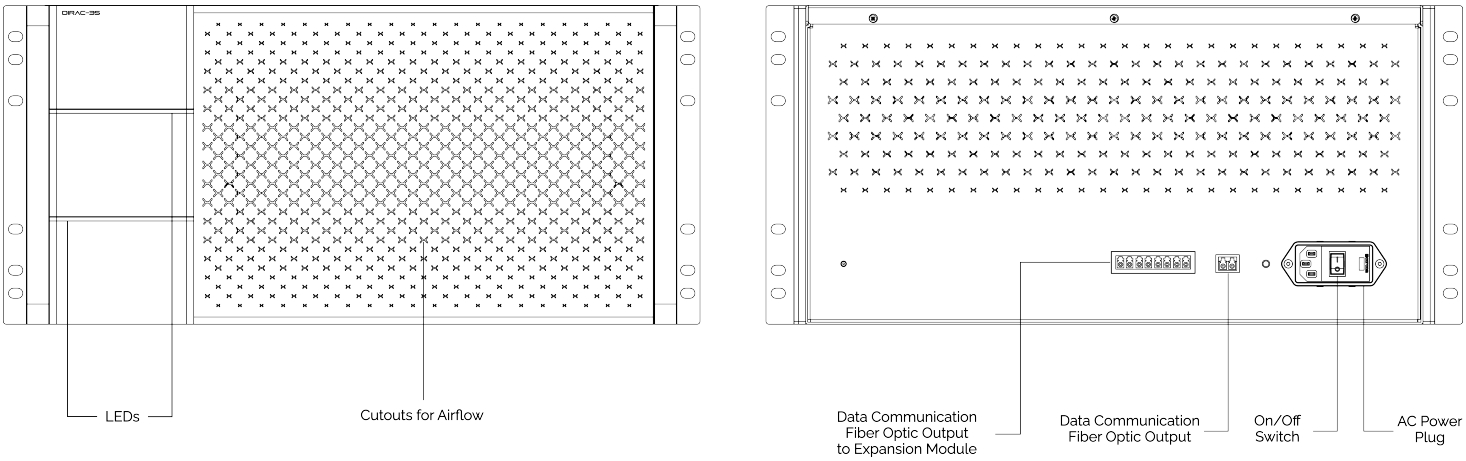

Figure 18: Front (left) and Back (right) panels of Dirac-3S Base Module.

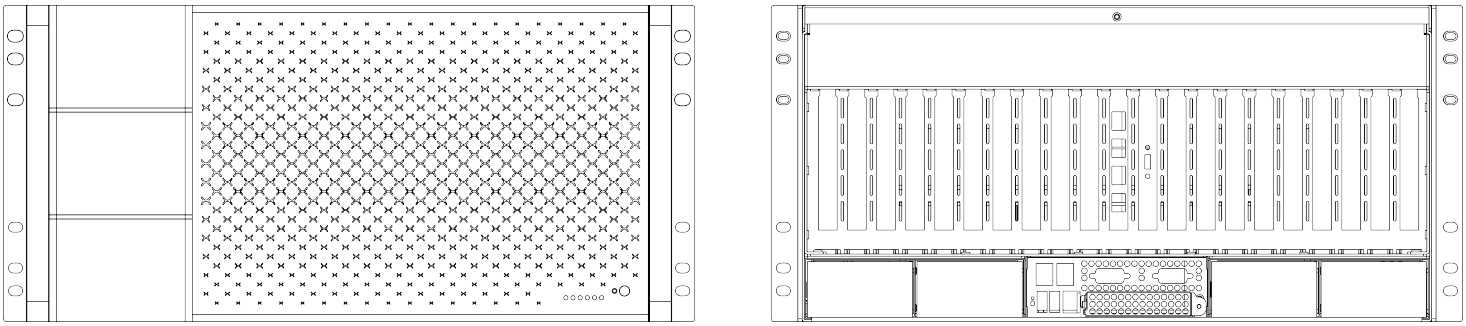

Figure 19: Front (left) and Back (right) panels of Dirac-3S Expansion Module.

6.2.5 Connecting Dirac-3 to a Host Computer

After powering on, ensure the device is idle and ready to connect via the blue-purple LEDs. The Dirac-3S device will be delivered with a static IP address of 192.168.42.13/24. To connect to it, configure your laptop's Ethernet adapter with an IP address in the 192.168.42.x range, where x is an unused address on the same subnet (e.g. not 13, 255, or 0) and the mask to 255.255.255.0. Once these settings have been changed, you may plug the Ethernet cable into the jack, and then into the host computer. You may now attempt to ping the device using the assigned IP to ensure connection to the device via the host computer:

```
ping 192.168.42.13
```

In order to verify that the Dirac server is accessible, users must utilize the system monitoring capabilities of the Python package eqc-direct. In order to install *eqc-direct* the latest available version from PyPi run:

```
pip install eqc-direct
```

The package supports Python versions 3.9 and 3.10. Next, using the Python package follow the instructions in the documentation under the System Monitoring heading to check if your device is up and running. When the device is powered on it begins by entering a calibration process to determine if the system is functioning properly. Using the system monitoring functions, users can check to see when their device is available to begin solving problems.

If needed, the MAC address for the device can be found on the documentation included upon arrival or by a sticker / panel on the device.

When the device is busy running computation, LEDs indicator becomes green. If the LED indicator becomes red, that means that Dirac-3 experience hardware failure or communication failure. Each gRPC request has a status and description in the response that can be used to verify any issues with the device. For additional information see Dirac-3S Software Interface Reference document.

LED color	Indication
Red	A hardware error has occurred
Blue-Violet	Device is idle
Green	Device is busy running a job
Bright Blue	Device is performing a calibration
Yellow	Device is performing a health check

Table 5: LEDs indication vs. Device status

Copyright 2026 Quantum Computing Inc.

7 Mechanical Specifications

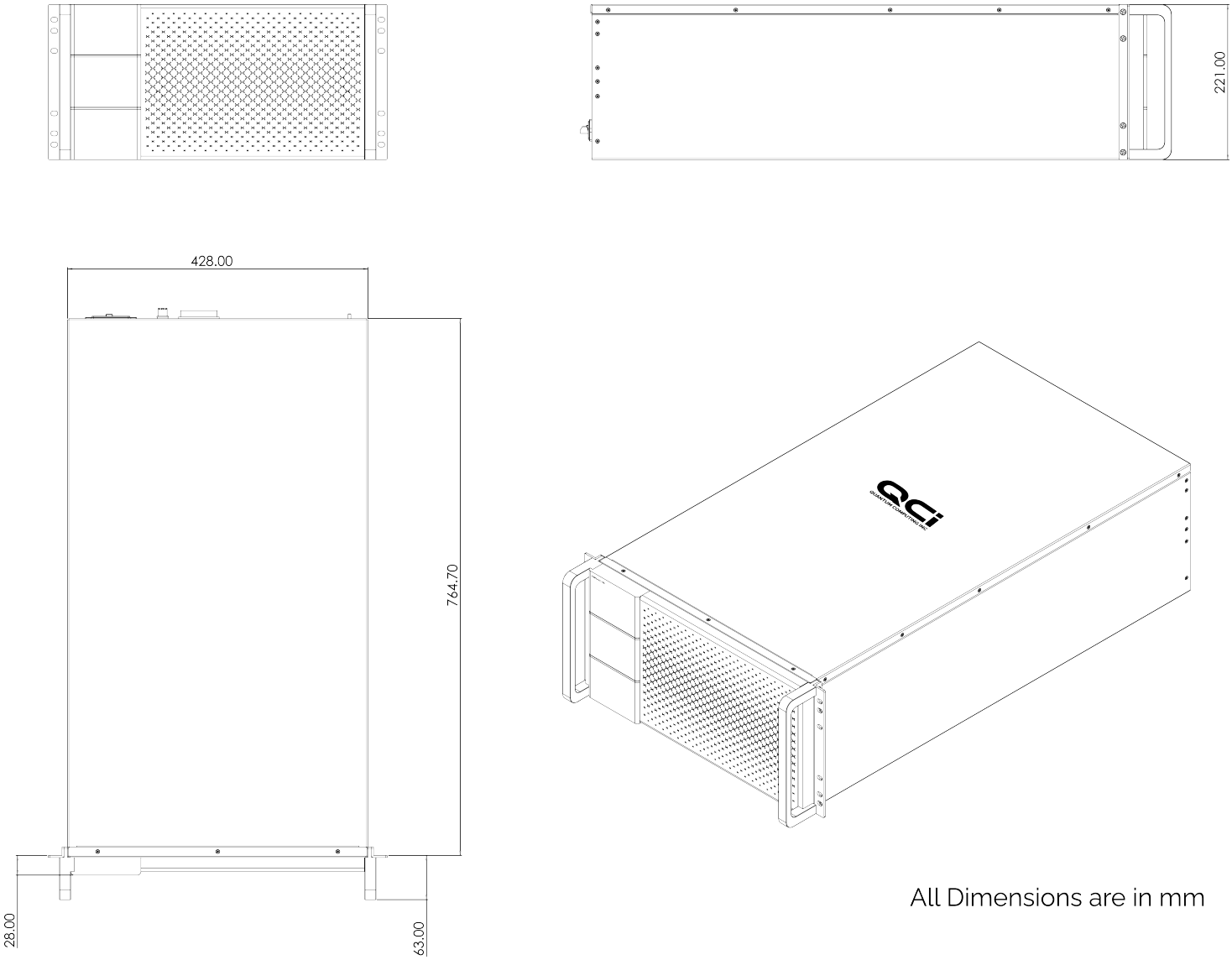

Figure 20: Dirac-3S Base Module

Copyright 2026 Quantum Computing Inc.

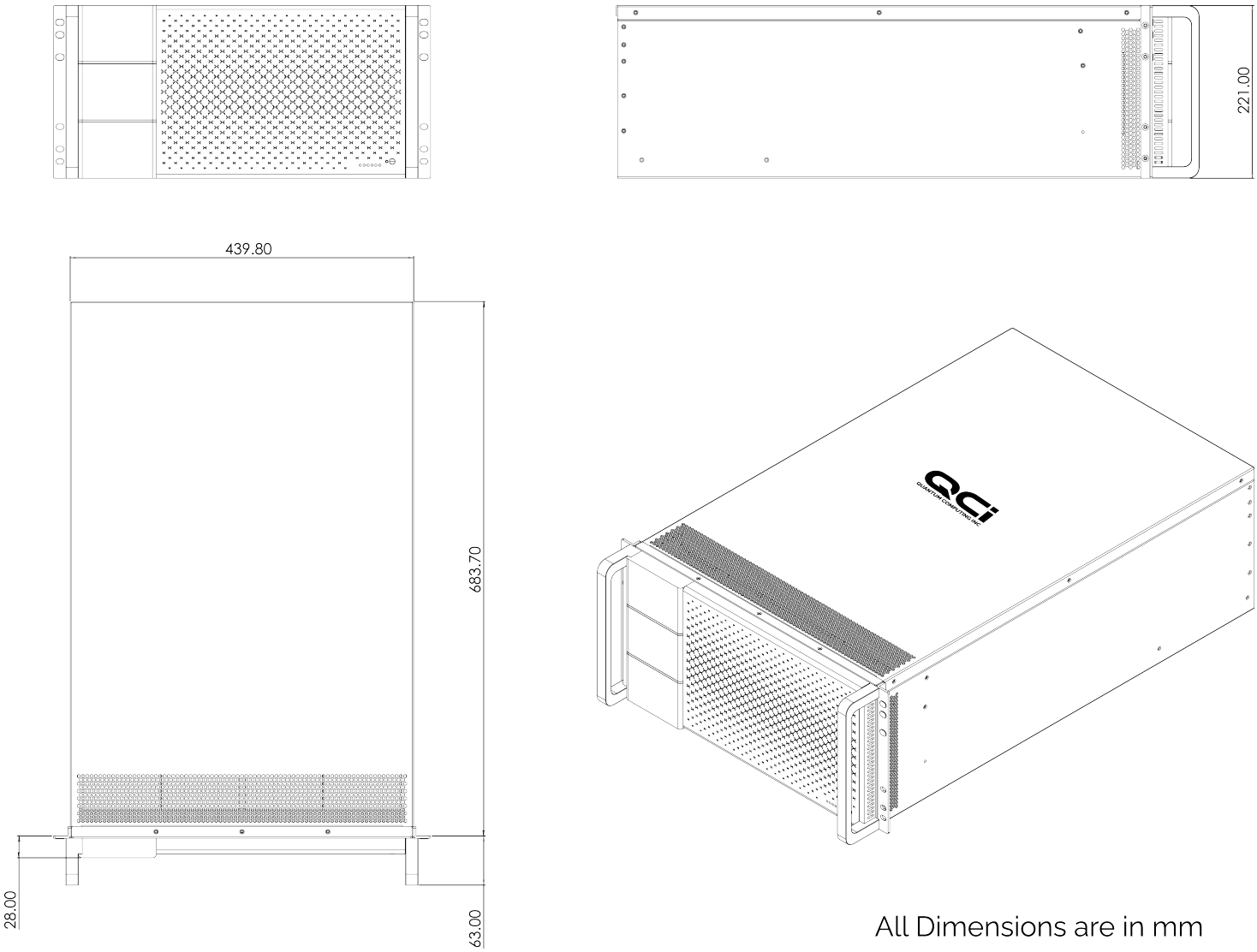

Figure 21: Dirac-3S Expansion Module

Copyright 2026 Quantum Computing Inc.

8 Related Technical References

The following documents provide additional technical details that complement this User Guide:

- **Dirac-3S Software Interface Reference** API descriptions, data formats, error messages, status, etc.
- **Dirac-3S Continuous Solver Benchmarking** Benchmark examples, test methodologies, performance metrics, and results.

9 Contacting Support

For assistance, please contact:
Email: support@quantumcomputinginc.com
Support Portal: https://quantumcomputinginc.ladesk.com/
When contacting support, provide:

- Dirac-3S serial number
- Software version
- Problem description
- Error messages
- Relevant log files

References

[1] Fabio Furini et al. "QPLIB: A Library of Quadratic Programming Instances". In: *Mathematical Programming Computation* (2018). DOI: 10.1007/s12532-018-0147-4.

[2] Nicolas Gisin et al. "Quantum cryptography". In: *Rev. Mod. Phys.* 74 (1 Mar. 2002), pp. 145–195. DOI: 10.1103/RevModPhys.74.145. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.145.

[3] A.D. King, J. Raymond, and T. et al. Lanting. "Quantum critical dynamics in a 5,000-qubit programmable spin glass". In: *Nature* 617 (2023), pp. 61–66. DOI: https://doi.org/10.1038/s41586-023-05867-2.

[4] Prajnesh Kumar et al. "Efficient reservoir computing using field programmable gate array and electro-optic modulation". In: *OSA Continuum* 4.3 (Mar. 2021), pp. 1086–1098. DOI: 10.1364/OSAC.417996. URL: https://opg.optica.org/osac/abstract.cfm?URI=osac-4-3-1086.

Copyright 2026 Quantum Computing Inc.

[5] Yi-Xuan Ma and Peng-Bo Li. "Deterministic generation of phononic Fock states via weak nonlinearities". In: *Phys. Rev. A* 108 (5 Nov. 2023), p. 053709. DOI: 10.1103/PhysRevA.108.053709. URL: https://link.aps.org/doi/10.1103/PhysRevA.108.053709.

[6] Lac Nguyen et al. *Entropy Computing: Entropy computing, a paradigm for optimization in open photonic systems*. 2025. URL: https://doi.org/10.1038/s42005-025-02324-6.

[7] M. Uria, P. Solano, and C. Hermann-Avigliano. "Deterministic Generation of Large Fock States". In: *Phys. Rev. Lett.* 125 (9 Aug. 2020), p. 093603. DOI: 10.1103/PhysRevLett.125.093603. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.093603.

[8] Y. Yamamoto, K. Aihara, and T. et al. Leleu. "Coherent Ising machines—optical neural networks operating at the quantum limit". In: *npj Quantum Inf* (2017). DOI: https://doi.org/10.1038/s41534-017-0048-9.

Copyright 2026 Quantum Computing Inc.