

Variational Inference for Quantum HyperNetworks

Luca Nepote^{*†}, Alix Lhéritier^{*}, Nicolas Bondoux^{*}, Marios Kountouris^{†‡} and Maurizio Filippone[§]

^{*}Amadeus, Sophia Antipolis, France

[†]EURECOM, Sophia Antipolis, France

[‡]Universidad de Granada, Spain

[§]KAUST, Statistics Program, Saudi Arabia

Abstract—Binary Neural Networks (BiNNs), which employ single-bit precision weights, have emerged as a promising solution to reduce memory usage and power consumption while maintaining competitive performance in large-scale systems. However, training BiNNs remains a significant challenge due to the limitations of conventional training algorithms. Quantum HyperNetworks offer a novel paradigm for enhancing the optimization of BiNN by leveraging quantum computing. Specifically, a Variational Quantum Algorithm is employed to generate binary weights through quantum circuit measurements, while key quantum phenomena such as superposition and entanglement facilitate the exploration of a broader solution space. In this work, we establish a connection between this approach and Bayesian inference by deriving the Evidence Lower Bound (ELBO), when direct access to the output distribution is available (i.e., in simulations), and introducing a surrogate ELBO based on the Maximum Mean Discrepancy (MMD) metric for scenarios involving implicit distributions, as commonly encountered in practice. Our experimental results demonstrate that the proposed methods outperform standard Maximum Likelihood Estimation (MLE), improving trainability and generalization.

Index Terms—Quantum HyperNetworks, Quantum Machine Learning, Variational Quantum Algorithm, Bayesian Inference, Binary Neural Networks

I. INTRODUCTION

Machine Learning (ML) is transforming industries and shaping daily life; however, the substantial computational power required to train and deploy advanced models leads to unsustainable energy consumption [1], [2]. As ML models continue to increase in complexity and scale, improving their energy efficiency has become a critical and urgent challenge.

To address this challenge, several techniques have been proposed, with quantization standing out as a particularly effective approach [3]. By reducing the precision of each model weight, quantization lowers memory usage requirements and speeds up computation, while maintaining competitive performance [4]. The most extreme form of quantization is *binarization*, where each parameter is constrained to a single bit representing values in $\{-1, +1\}$. *Binary Neural Networks* (BiNNs) [5], [6] leverage this principle to drastically reduce energy consumption and memory usage. When scaled appropriately, BiNNs can achieve performance comparable to their floating-point (full-precision) counterparts [7], making them a promising solution for efficient deep learning models at scale.

However, the discrete nature of binary weights poses significant challenges for optimization. Traditional deep learning training techniques, which rely on continuous weight updates, are not well-suited to the constraints imposed by BiNNs. As a result, alternative training strategies are necessary to overcome these limitations, requiring the development of novel optimization methods for binary-weight networks [8]. A promising approach for training such models lies in the use of Quantum Computing. When a quantum circuit is measured, the resulting bitstrings can be directly employed as binary weights in a BiNN, effectively addressing the limitations of single-bit precision and eliminating the need for backpropagation. Furthermore, by leveraging fundamental quantum phenomena such as *superposition* and *entanglement*, it becomes possible to explore a larger solution search space in quantum superposition, potentially leading to more efficient and effective optimization.

This work investigates and advances a promising approach known as *Quantum HyperNetworks* [9]: a training technique at the intersection of quantum computing and machine learning. Specifically, we focus on *Quantum Machine Learning* (QML) and *Variational Quantum Algorithms* (VQAs) [10] to explore how quantum computing can be integrated into deep learning systems for more efficient and scalable training.

As highlighted in [9], Quantum HyperNetworks are inherently linked to the Bayesian approach, as their output defines a probability distribution over the weights of a BiNN. The core contribution of our work is the derivation of the Evidence Lower Bound (ELBO) for Quantum HyperNetworks, along with a surrogate version tailored for practical implementation. These formulations introduce an implicit form of regularization, thereby enhancing both trainability and generalization. While prior studies [11], [12] have already studied the use of distance metrics such as Kullback-Leibler (KL) divergence and the Maximum Mean Discrepancy (MMD) in quantum generative modeling - focusing on data generation and the representation power of quantum circuits - our work applies variational inference [13] to derive regularization terms based on KL and MMD within the training loss. We analyze the influence of these terms on the optimization process, highlighting their role in enhancing performance and stability.

Correspondence to: Luca Nepote <luca.nepote@eurecom.fr>, Alix Lhéritier <alix.lheritier@amadeus.com>.

II. PRELIMINARIES

As previously mentioned, *Binary Neural Networks* offer a promising solution for both energy- and memory-limited devices. However, a major challenge in training these models lies in the nested optimization loop required to update parameters, hyperparameters, and architectural choices. This process is computationally intensive due to the presence of both inner and outer optimization loops, resulting in a significant computational cost and an unsustainable carbon footprint, similar to their floating-point counterparts. Indeed, even with BiNNs, floating-point numbers are required for the parameter updates during training [6].

In [9], the authors introduced *Quantum HyperNetworks* as a novel approach to address this challenge. A *HyperNetwork* [14] is a machine learning technique in which one network generates the weights for another, typically larger, network, accelerating the search within high-dimensional parameter spaces containing millions of variables [15]. This study explored a quantum-inspired strategy that jointly optimizes all the *augmented parameters*—i.e. parameters, hyperparameters, and architectural choices—within a single optimization loop by leveraging quantum states. By exploiting key quantum properties such as superposition and entanglement, the approach enables more efficient exploration of the solution space, reducing computational overhead.

A. VQA for Quantum HyperNetworks

Our objective is to identify ideally optimal solutions for the classical supervised ML problem, based on a Neural Network function $\text{NN}(\mathbf{x}; \mathbf{w})$, with input \mathbf{x} and augmented parameters $\mathbf{w} \in \{-1, +1\}^N$ with the following cost function

$$C(\sigma) = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell(\text{NN}(\mathbf{x}_i; 2\sigma - 1), \mathbf{y}_i) \quad (1)$$

where N_s denotes the training set size, \mathbf{x}_i represents an input sample with the corresponding label \mathbf{y}_i in the dataset $(Y; X)$, with $X = \{\mathbf{x}_i\}_{i=1}^{N_s}$ and $Y = \{\mathbf{y}_i\}_{i=1}^{N_s}$, and ℓ is the loss function for the supervised ML problem. To encode the problem into a quantum state suitable for a VQA, each parameter is mapped onto one of N qubits in the computational basis,

$$|\Psi\rangle = \sum_{\sigma_1, \dots, \sigma_N} \Psi(\sigma_1, \dots, \sigma_N) |\sigma_1, \dots, \sigma_N\rangle, \quad (2)$$

with $\sigma_i \in \{0, 1\}$. One way to build a Hamiltonian from the classical cost function is to consider the Pauli operator Z acting on the i -th qubit, i.e.

$$\hat{\sigma}_i^z \equiv I^{\otimes i-1} \otimes Z \otimes I^{\otimes n-i}, \quad (3)$$

with the eigenvalue equation

$$\hat{\sigma}_i^z |\sigma_1, \dots, \sigma_i, \dots, \sigma_N\rangle = (2\sigma_i - 1) |\sigma_1, \dots, \sigma_i, \dots, \sigma_N\rangle. \quad (4)$$

Let us consider $w_i \equiv 2\sigma_i - 1$. By noting that any Boolean function $C(\sigma_1, \dots, \sigma_N)$ can be rewritten as a multilinear combination of the variables w_i and applying the same multilinear

combination to the $\hat{\sigma}_i^z$ operators, the following $2^N \times 2^N$ diagonal Hamiltonian is obtained

$$\hat{C} = \begin{pmatrix} C(\sigma_1) & 0 & \cdots & 0 \\ 0 & C(\sigma_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C(\sigma_{2^N}) \end{pmatrix}, \quad (5)$$

where each diagonal element of this operator represents the cost value associated with a specific BiNN configuration σ_i , resulting in a total of 2^N possible BiNNs. Notably, other encodings are possible, enabling the use of alternative computational bases to suit different optimization strategies.

Each basis element corresponds to a specific configuration of augmented parameters, encompassing all model parameters, hyperparameters, and architectural choices of the BiNN. Any additional decision is encoded using an extra qubit: for instance, if the choice involves selecting between two activation functions a_1 and a_2 , an additional qubit σ is introduced, and the activation is determined based on its value

$$a(\mathbf{x}; \sigma) = \begin{cases} a_1(\mathbf{x}) & \text{if } \sigma = 0, \\ a_2(\mathbf{x}) & \text{if } \sigma = 1. \end{cases} \quad (6)$$

It is important to note that, unlike the previous discussion on BiNNs, activations in this approach are not restricted to binary values—only the weights are binary.

When the quantum state is measured, it yields the augmented parameters for the BiNN. This encoding is well-suited for a VQA, where, after training, the system is expected to return an optimized Neural Network (NN) configuration.

The quantum state $|\Psi\rangle$ is obtained by a parametrized quantum circuit $U(\theta)$, such that

$$|\Psi_\theta\rangle = U(\theta)|0\rangle^{\otimes n}. \quad (7)$$

The quantum circuit is constructed as a sequence of L unitary blocks, that is, $U(\theta) = U_L(\theta_L) \dots U_1(\theta_1)$, where each block consists of a set of linear operations, namely CX , RY , and RZ gates. The circuit depth is adjustable, allowing control over the degree of entanglement within the quantum system. Each set of unitary operations is expressed as

$$U_k(\theta_k) = \prod_{\substack{m=1+k \text{ mod } 2 \\ \text{step } 2}}^{N-1} CX(m, m+1) \times \prod_{j=1}^N RY(j, \theta_{0,j,k}) RZ(j, \theta_{1,j,k}). \quad (8)$$

The observable is defined through a stochastic relaxation of the discrete optimization problem, enabling the search for an optimal solution in the augmented parameter distribution. Mathematically, the expectation value of the Hamiltonian \hat{C} —the energy E —is defined as:

$$E(\theta) = \langle \Psi_\theta | \hat{C} | \Psi_\theta \rangle \quad (9)$$

and the best approximation to the problem is given by

$$\theta^* = \arg \min_{\theta} E(\theta), \quad (10)$$

since $\min_{\theta} E(\theta) \geq \lambda^*$, λ^* being the lowest eigenvalue of \hat{C} , with equality if there exists θ^* such that $|\Psi_{\theta^*}\rangle$ is equal to the eigenvector corresponding to λ^* .

B. Optimization

For the optimization, the standard gradient descent method is used in [9]. The expected value $E(\theta)$ and the gradient $\nabla_{\theta} E(\theta)$ are estimated through statistical averages over multiple quantum circuit measurements. These values are then processed by a classical optimizer to iteratively update the parameters until convergence. Consider

$$\begin{aligned} E(\theta) &= \langle \Psi_{\theta} | \hat{C} | \Psi_{\theta} \rangle \\ &= \sum_{\sigma_1, \sigma_2, \dots, \sigma_N} |\Psi_{\theta}(\sigma_1, \sigma_2, \dots, \sigma_N)|^2 C(\sigma_1, \sigma_2, \dots, \sigma_N) \\ &= \mathbb{E}_{\tilde{\sigma} \sim |\Psi_{\theta}|^2} [C(\tilde{\sigma})] \approx \frac{1}{N_{qc}} \sum_{i=1}^{N_{qc}} C(\tilde{\sigma}_i) \end{aligned} \quad (11)$$

where $\tilde{\sigma}_i$ is one of the N_{qc} sampled BiNN configurations from the quantum circuit. Considering this Monte-Carlo method is crucial because, if an exhaustive search of all possible BiNN configurations was conducted to construct the Hamiltonian, the optimal solution would already be known before running the VQA - making the quantum algorithm unnecessary [16].

In a real experimental setting, the *parameter shift rule* [17] is used to compute gradients efficiently:

$$\frac{\partial E(\theta)}{\partial \theta_{\alpha, j, k}} = \frac{1}{2} [E(\theta_{\alpha, j, k}^+) - E(\theta_{\alpha, j, k}^-)] \quad (12)$$

where the elements of the shifted parameter vector $\theta_{\alpha, j, l}^{\pm}$ are

$$\theta_{\beta, m, l}^{\pm} = \theta_{\beta, m, l} \pm \frac{\pi}{2} \delta_{\alpha, \beta} \delta_{m, j} \delta_{k, l}. \quad (13)$$

The expected values for the gradient are estimated by a Monte Carlo approach for each shifted parameter. Previous work [9] uses a *Tensor Network* [18] simulator combined with automatic differentiation, employing the L-BFGS algorithm for optimization.

III. PROPOSED METHODOLOGY

In our implementation, we design a circuit with linear connectivity following (8), and perform measurements in the Pauli Z basis. We initialize the system in the zero state vector but instead of directly constructing the Hamiltonian of (5), we employ (11) with the parameter shift rule of (12). This ensures compliance with practical quantum settings while maintaining computational efficiency.

As noted by the authors in [9], the Quantum HyperNetworks approach established a direct connection to Bayesian inference. Specifically, the quantum circuit $|\Psi_{\theta}\rangle$ defines a probability distribution over the binary weights of the BiNN. While direct evaluation of the posterior distribution is intractable, we can approximate it using Variational Inference (VI) by estimating the ELBO. We consider here two cases: (i) we assume full access to the quantum circuit distribution, such as in simulations where the amplitudes and corresponding

probabilities of each bitstring are known; (ii) we adopt a more realistic setting in which the quantum circuit distribution is implicit and only accessible through samples (i.e., quantum measurements).

A. Explicit ELBO

Following a VI approach, we can approximate the true posterior distribution over binary weights using a parametric family $q_{\theta}(\sigma)$, where θ represents the variational parameters. In our case, θ corresponds to the rotational angles of the RZ and RY gates in the quantum circuit. The probability distribution over the binary weights is governed by the *Born rule*

$$q_{\theta}(\sigma) = |\langle \sigma | U(\theta) | 0 \rangle|^2, \quad (14)$$

which defines the probability of measuring each bitstring as a function of the variational parameters θ .

Considering the general ELBO expression

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_{\theta}(\sigma)} [\log p(Y | X, \sigma)] - \text{KL}[q_{\theta}(\sigma) \| p(\sigma)], \quad (15)$$

the first term is the model fitting term (expected log-likelihood), which we approximate using Monte Carlo sampling by measuring the quantum circuit. The estimator is unbiased, with variance scaling as $1/N_{qc}$, where N_{qc} denotes the number of measurements. In the context of a classification problem, X corresponds to the set of data points with labels Y . The second term serves as a penalty or regularization term, corresponding to the KL divergence between the variational distribution and the prior distribution over the weights of the BiNN. For this derivation, we impose a uniform prior for each parameter $\{0, 1\}$:

$$\sigma_i \sim \mathcal{U}\{0, 1\}, \quad (16)$$

with σ_i being the i -th component of the bitstring σ .

Extending this to all binary weights, we obtain a joint uniform distribution over all possible BiNN configurations. This implies that each configuration has an equal probability of being sampled, specifically $1/N_t$, where $N_t = 2^{n_{\text{qubits}}}$ represents the total number of possible configurations.

Given the uniform prior p , we derive the following expression for the regularization term in (15):

$$\begin{aligned} \text{KL}[q_{\theta}(\sigma) \| p(\sigma)] &= \sum_{\sigma} q_{\theta}(\sigma) \log \frac{q_{\theta}(\sigma)}{p(\sigma)} \\ &= \sum_{\sigma} q_{\theta}(\sigma) \log q_{\theta}(\sigma) - \sum_{\sigma} q_{\theta}(\sigma) \log p(\sigma) \\ &= \sum_{\sigma} q_{\theta}(\sigma) \log q_{\theta}(\sigma) - \sum_{\sigma} q_{\theta}(\sigma) \log \frac{1}{N_t} \\ &= \sum_{\sigma} q_{\theta}(\sigma) \log q_{\theta}(\sigma) + \log N_t, \end{aligned} \quad (17)$$

where the first term corresponds to the negative entropy of the quantum circuit variational distribution, and the second term is a constant that does not influence the optimization process.

By substituting this term into (15), we obtain the final expression to be maximized

$$\mathcal{L}_{\text{ELBO}} \stackrel{c}{=} \mathbb{E}_{q_{\theta}(\sigma)} [\log p(Y | X, \sigma)] - \mathbb{E}_{q_{\theta}(\sigma)} [\log q_{\theta}(\sigma)], \quad (18)$$

where $\stackrel{c}{=}$ denotes equality up to a constant term. Note that the second term of (18) is the Shannon entropy of q_θ .

In experimental settings, accurately estimating the entropy of the output distribution requires collecting multiple samples, which can be computationally expensive. In contrast, simulations may provide direct access to the full state vector, allowing precise computation of the probability distribution over bitstrings.

B. Surrogate ELBO

In this subsection, we derive a *Surrogate Evidence Lower Bound* (SELBO) for implicit distributions. This approach is applicable to both simulations and real quantum hardware, where direct access to the quantum state distribution is unavailable, and only sampled data can be used. The challenge of applying variational inference to implicit distributions has also been explored in [19], where the authors consider semi-implicit distributions, constructed as a mixture of explicit conditional distributions.

We adopt the approach proposed in [20], where the KL divergence regularization term is replaced with MMD [21]. MMD is an integral probability metric [22] that compares two probability distributions using sample data rather than their explicit distributions. In our setting, one of the distributions is the implicit distribution $q_\theta(\sigma)$ over the binary weights, defined by the quantum circuit and accessible only through measurements. The other is the uniform prior $p(\sigma)$, from which we assume to have only a finite set of samples during computation. Then, given two sets of samples $\{x_i\}_{i=1}^n \sim q_\theta(\sigma)$ and $\{y_j\}_{j=1}^m \sim p(\sigma)$,

$$\begin{aligned} \text{MMD}^2(q_\theta(\sigma), p(\sigma)) &:= \\ \mathbb{E}_{x, x' \sim q_\theta} [k(x, x')] + \mathbb{E}_{y, y' \sim p} [k(y, y')] - 2\mathbb{E}_{\substack{x \sim q_\theta \\ y \sim p}} [k(x, y)], \end{aligned} \quad (19)$$

where $k(\cdot, \cdot)$ is a positive-definite kernel, which we assume to be the Gaussian radial basis function (RBF) kernel, i.e.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{h^2}\right), \quad (20)$$

with h being the kernel bandwidth parameter. This parameter determines the scale of the Gaussian kernel: a small h results in a narrow, localized kernel that closely approximates a Dirac delta function. This enhances sensitivity to local differences (low bias) between samples of the two distributions, but makes the estimation more susceptible to noise (high variance). Conversely, a large h produces a broader, smoother kernel that reduces sensitivity to local variations (low variance) while emphasizing global differences. However, this comes at the cost/risk of higher bias, potentially overlooking important or finer details. Thus, choosing an appropriate h is essential to effectively compare distributions, ensuring a balance between capturing local variations and preserving global structures.

We consider the squared MMD and the following unbiased estimator [21]

$$\begin{aligned} \widehat{\text{MMD}}^2_{\text{U}} &= \frac{1}{n(n-1)} \sum_{\substack{i, j=1 \\ i \neq j}}^n k(x_i, x_j) \\ &\quad + \frac{1}{m(m-1)} \sum_{\substack{i, j=1 \\ i \neq j}}^m k(y_i, y_j) \\ &\quad - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x_i, y_j). \end{aligned} \quad (21)$$

To derive a *surrogate* version of the ELBO for the Quantum HyperNetworks problem, we replace the KL divergence term with the MMD term, scaled by a factor λ :

$$\mathcal{L}_{\text{SELBO}} = \mathbb{E}_{q_\theta(\sigma)} [\log p(Y | X, \sigma)] - \lambda \text{MMD}^2(q_\theta(\sigma), p(\sigma)). \quad (22)$$

The scaling factor λ determines the amount of regularization during optimization. Different choices for λ are possible, and the optimal one is highly problem-dependent. Annealing schedules are also possible [20], [23], to gradually transition towards *Maximum Likelihood Estimation* (MLE) as training progresses. In our study, we consider λ as a constant term, and we leverage the regularization term primarily to enhance the trainability of the Quantum HyperNetwork and to analyze its impact on the loss landscape.

IV. EXPERIMENTS

We evaluate our approach using three different toy datasets, illustrated in Fig. 2. The first dataset (a), taken from [9], consists of a two-dimensional distribution with four Gaussian clusters. The other two datasets, (b) and (c), correspond to 2D Moon and Ring shapes, respectively. All three datasets are balanced, with each of the two classes containing 150 data points for training and 100 samples for testing. In all cases, we consider a binary classification problem, where the objective is to accurately predict the class of each 2D data point.

To perform classification, we use a simple BiNN consisting of a single hidden layer with 3 neurons and a total of 14 binary parameters, similarly to [9]. Note that the value of the last bit in the measured bitstring, denoted as σ_N , determines the activation function for the hidden layer: if $\sigma_N = 0$, ReLU is used, otherwise Sigmoid is applied. The output layer uses a fixed Sigmoid activation function.

For the optimization task, we used gradient ascent with a variable learning rate, initially set to a high value of $\eta = 1$: if the improvement in training loss falls below a threshold $\min \delta$ for a given number of consecutive epochs (defined by the *Patience* parameter), the learning rate is reduced by a predefined *Decay Factor*. This adaptive process continues until the total number of epochs, n_{epochs} , is reached. A high initial learning rate is essential, as noted in [7], since the low precision of binary network weights means that small parameter updates often yield negligible performance improvements. All algorithmic specifications are detailed in Table I.

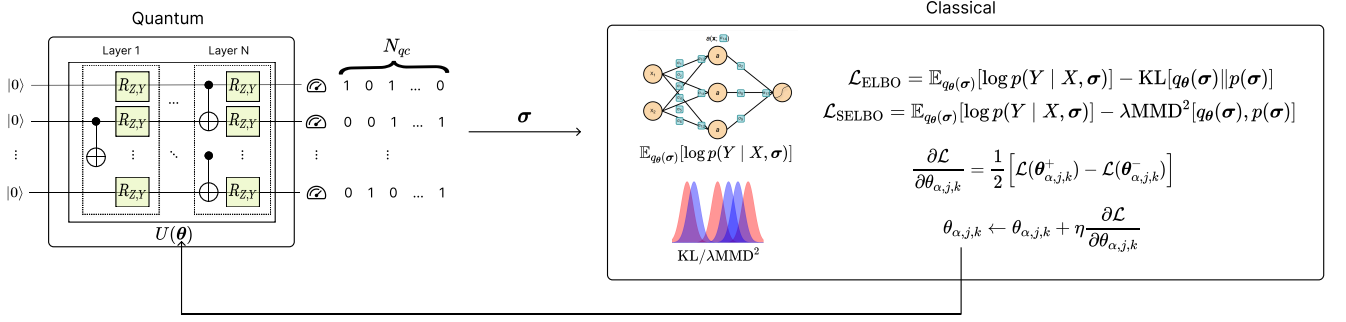


Fig. 1. Representation of the Regularized Quantum HyperNetworks algorithm.

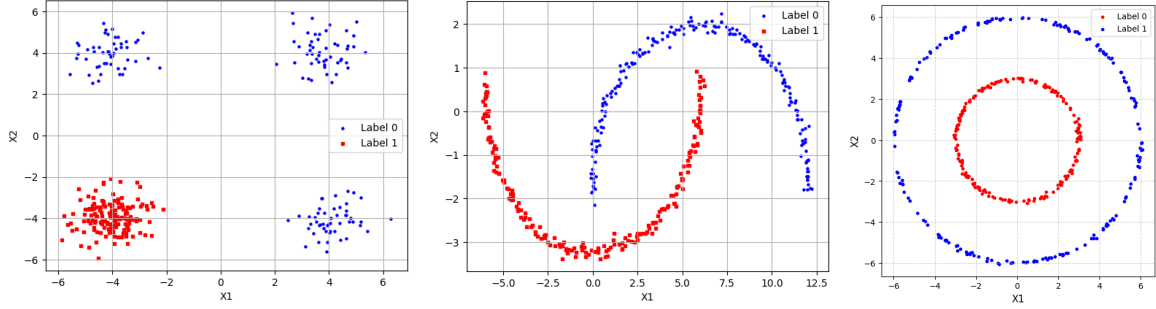


Fig. 2. Datasets used in the experiments: 2D Gaussian dataset (a), 2D Moon dataset (b), 2D Ring dataset (c).

TABLE I
TRAINING PARAMETERS

Parameter	Value
Number of Epochs (n_{epochs})	200
Patience	3
Minimum Delta (min δ)	10^{-4}
Decay Factor	0.5
Learning Rate (η)	1
Number of Keys (n_{keys})	100

We randomly initialize the gate parameters of the quantum circuit using a uniform distribution over the interval $[0, 2\pi]$, and repeat the optimization process across 100 different initializations to ensure statistical significance.

In our experiments using SELBO, we set the bandwidth parameter to $h = \frac{n_{\text{qubits}}}{4}$, following the considerations outlined in [24]. This choice links the bandwidth to the number of qubits in the system, facilitating comparisons within a continuous embedding space of discrete samples. The underlying motivation is that the separation between samples is determined by the value of each qubit, which is 1, making this scaling choice essential for maintaining consistency across different configurations.

The procedure in Algorithm 1 is executed using the *qujax* library [25], a quantum software framework designed for efficient quantum circuit simulation on GPUs. This simulation provides direct access to the full state-vector representation, including the corresponding amplitudes and probabilities of all possible outcomes. As a result, we can compute the entropy term in the explicit ELBO formulation without relying on re-

peated measurements to approximate the bitstring distribution, thus optimizing the loss function in (18).

Algorithm 1 Regularized Quantum HyperNetworks

- 1: **Input:** Initial quantum circuit parameters θ , number of epochs n_{epochs} , input data
- 2: **Output:** Optimized parameters θ^*
- 3: $\theta^* \leftarrow \theta$, $\mathcal{L}^* \leftarrow \text{SELBO}(\theta)$
- 4: **for** $i \leftarrow 1$ to n_{epochs} **do**
- 5: Compute gradients using the parameter-shift rule:

$$\frac{\partial \mathcal{L}}{\partial \theta_{\alpha,j,k}} = \frac{\text{SELBO}(\theta_{\alpha,j,k}^+) - \text{SELBO}(\theta_{\alpha,j,k}^-)}{2}$$

- 6: Update parameters using gradient ascent:

$$\theta_{\alpha,j,k} \leftarrow \theta_{\alpha,j,k} + \eta \frac{\partial \mathcal{L}}{\partial \theta_{\alpha,j,k}}$$

- 7: **if** $\text{SELBO}(\theta) > \mathcal{L}^*$ **then**
- 8: $\theta^* \leftarrow [\theta_{\alpha,j,k}]$, $\mathcal{L}^* \leftarrow \text{SELBO}(\theta)$
- 9: **end if**
- 10: **end for**
- 11: **return** θ^*
- 12: **function** $\text{SELBO}(\theta)$
- 13: Measure N_{qc} bitstrings from the quantum circuit $|\Psi_{\theta}\rangle$
- 14: Upload the binary weights into the BiNN
- 15: **return** $\mathcal{L}_{(\text{S})\text{ELBO}}(\theta)$
- 16: **end function**

We compare the performance of our proposed methods, based on the ELBO and its implicit variant (SELBO), with

results obtained using MLE, as employed in previous work [16]. All experiments are conducted using quantum circuits with $N_{\text{layers}} = 1$ and a fixed number of measurements $N_{\text{qc}} = 100$. We opt for a shallow circuit depth because a low amount of entanglement is sufficient for the specific task at hand [9]. In fact, the best solutions are often obtained with circuits containing fewer layers, which also makes them more amenable to classical simulation in small-scale settings. Notice that the circuit parameters are initialized identically across all three methods to ensure a fair comparison.

In Fig. 3, we report the ELBO curves for the SELBO and ELBO methods on the Gaussian dataset. We represent the average behavior over the 100 initializations (random seeds), where we show the mean value and the standard deviation over the different initializations. We can see how the expected log-likelihood and the regularization term are correctly increasing, both in the explicit and implicit cases, meaning that we are moving away from the uniform prior.

In Fig. 4 we show the training curves for different methods (ELBO, SELBO and MLE) across the three datasets. We can see that the (S)ELBO values for $\lambda \in \{0.01, 1\}$ go above the MLE in spite of the penalization, thus indicating a better trainability. Moreover, it seems to help for the optimization task, intuitively smoothing the high-dimensional search space and helping it to escape from local optima. Note that the amount of regularization is also an important factor: considering high value of λ the SELBO diverges from the original ELBO, possibly degrading the performance. However, from the experimental results, we notice that there is a range of λ values that can be considered to achieve good performance for the classification task, whereas for higher values (i.e. from $\lambda = 1000$) the regularization overcomes the expected log-likelihood, leading to bad binary configurations.

In Table II-Table III, we report the final *binary cross entropy* (BCE) losses and accuracies for each dataset on the test set, trained with the three different methods. Note that, for the SELBO, we consider a constant $\lambda \in \{0.01, 1.0, 100.0\}$. The “MLE E.S.” in the tables refers to the test loss considering the best binary configuration on the training set, trained with MLE via Exhaustive Search, that is the brute force approach trying all the possible $2^{N_{\text{qubits}}}$ configurations. “MLE” in the tables refers to the quantum circuit approach of [16], using gradient ascent and Monte Carlo sampling.

On average, the SELBO outperforms the standard MLE method. We observe that, in most cases, the (S)ELBO leads to a higher and more concentrated accuracy. The best solution is found with the SELBO method and $\lambda = 0.01$, meaning that only a small amount of regularization is needed to achieve better solutions. However, we note that for the Rings dataset, the problem is generally more challenging to solve due to the limitations of the simple binary neural network under study.

We also acknowledge that, thanks to the regularization term in the VI formulation, one can also reach better performances than those obtained with “MLE E.S.”. This is reasonable because the optimal solution on the training set may not necessarily be the best on the test set. Instead, the

Quantum HyperNetworks approach overcomes this issue by leveraging the uncertainty in sampling binary configurations: this is achieved through the probabilistic distribution, which allows for a more robust exploration of possible configurations, improving generalization beyond the training set.

Finally, to better understand the effect of regularization on the optimization process, we plot the evolution of the average gradient magnitude during training for the three different datasets in Fig. 5.

One can see that, when the training curves in Fig. 4 start to diverge, the gradients for the MLE case are higher and less concentrated than those employing the (S)ELBO. Additionally, one can note that for the Rings dataset, even if the problem is hard to solve for the simple BiNN under consideration, the (S)ELBO gradients are still slightly better than those obtained with MLE.

We also illustrate in Fig. 6 the loss landscapes around a locally optimal point for the Gaussian dataset, comparing the three different methods. It can be observed that regularization smooths the landscape, thereby facilitating optimization and enhancing trainability.

Regarding the necessary computing resources to run the experiments, a typical run on these toy datasets takes about 55 minutes on an NVIDIA Tesla T4 GPU, consuming about 150 kJ with approximately 12 GB of peak GPU memory usage. Given these high computing requirements, it is challenging to perform simulations with larger datasets and networks.

V. CONCLUSIONS AND FUTURE WORK

In this work, we established a connection between *Quantum HyperNetworks* and Bayesian inference by deriving the *Evidence Lower Bound* (ELBO) for optimizing *Binary Neural Networks* using quantum circuits. We introduced two formulations: an *explicit* ELBO, which leverages direct access to the quantum circuit distribution, and a *surrogate* ELBO, tailored for scenarios where only implicit distributions are accessible, as commonly encountered in practical quantum hardware implementations.

Our results show that optimizing BiNNs with ELBO-based variational methods enhances both trainability and generalization compared to conventional Maximum Likelihood Estimation. The ELBO framework facilitates better parameter updates by smoothing the loss landscape, leading to more efficient and generalized training. Moreover, this method could mitigate possible overfitting issues in complex training scenarios, thanks to the employment of the regularization term.

Our findings underscore the benefits of using an ELBO-driven approach for BiNN training and suggest that quantum-inspired variational methods may offer a principled pathway toward more effective learning. Future work could further explore the theoretical relationship between explicit and surrogate ELBO formulations, particularly through the KALE divergence [26]. Additionally, improvements in parameter initialization for quantum circuits could improve convergence stability and training efficiency. Empirical validation on real quantum hardware will be crucial in assessing the practical

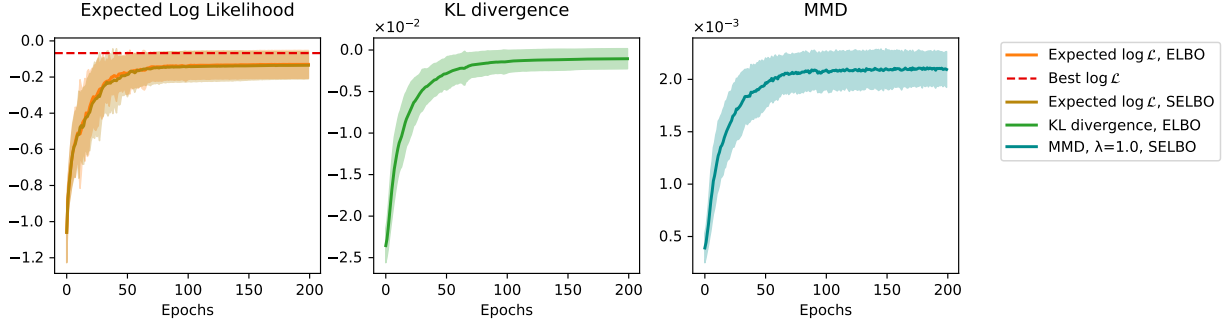


Fig. 3. Average (S)ELBO for $N_{\text{layers}} = 1$, $N_{\text{qc}} = 100$, run for 100 different initializations. Gaussian dataset. The KL is represented up to the constant term.

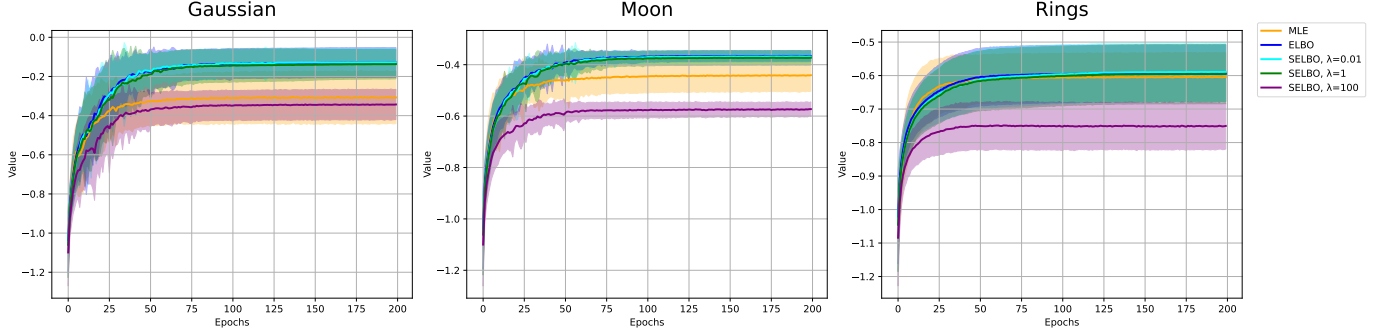


Fig. 4. Training curves for (S)ELBO and MLE, $N_{\text{layers}} = 1$, $N_{\text{qc}} = 100$, 100 different initializations.

TABLE II
FINAL BCE LOSS COMPARISON BETWEEN MLE, ELBO, AND SELBO ON THE TEST SETS. THE LOWER THE BETTER.

Dataset	MLE E.S.	MLE	ELBO	SELBO ($\lambda = 0.01$)	SELBO ($\lambda = 1.0$)	SELBO ($\lambda = 100.0$)
GAUSSIAN	0.067	0.315 ± 0.123	0.139 ± 0.073	0.129 ± 0.071	0.137 ± 0.074	0.141 ± 0.083
MOON	0.357	0.450 ± 0.062	0.378 ± 0.024	0.371 ± 0.022	0.374 ± 0.027	0.387 ± 0.033
RINGS	0.520	0.612 ± 0.070	0.608 ± 0.085	0.599 ± 0.084	0.603 ± 0.086	0.622 ± 0.085

TABLE III
ACCURACY COMPARISON BETWEEN MLE, ELBO, AND SELBO ON THE TEST SETS. THE HIGHER THE BETTER.

Dataset	MLE E.S.	MLE	ELBO	SELBO ($\lambda = 0.01$)	SELBO ($\lambda = 1.0$)	SELBO ($\lambda = 100.0$)
GAUSSIAN	0.993	0.893 ± 0.108	0.986 ± 0.012	0.991 ± 0.011	0.990 ± 0.012	0.987 ± 0.019
MOON	0.810	0.766 ± 0.082	0.808 ± 0.008	0.810 ± 0.006	0.809 ± 0.004	0.806 ± 0.008
RINGS	0.855	0.698 ± 0.101	0.698 ± 0.141	0.712 ± 0.145	0.707 ± 0.144	0.671 ± 0.137

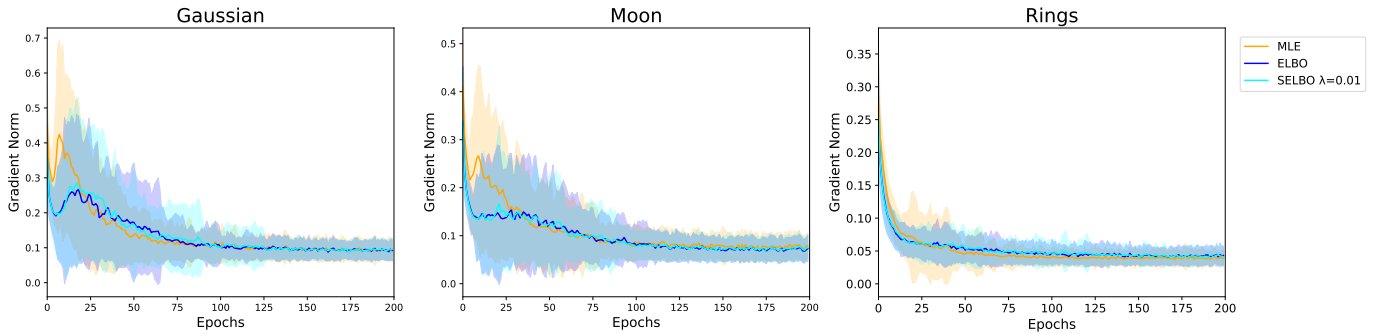


Fig. 5. Evolution of the average gradient magnitude during training.

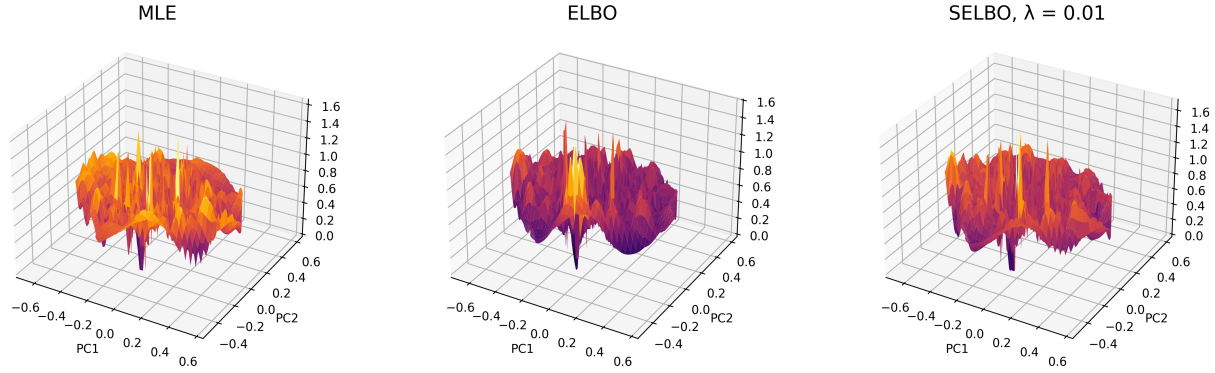


Fig. 6. Loss landscapes for different optimization procedures. Gaussian dataset.

feasibility of the proposed approach, particularly in a noisy environment. Furthermore, investigating the scalability of this method, especially in the context of *Multi-Basis Encoding* [16], [27], will be essential for larger networks, as well as to extend this framework to support higher precision weights [4]. Lastly, since training multiple binary weights using a single qubit can potentially introduce performance degradation, developing effective mitigation strategies will be key to ensuring robust performance at larger scales.

REFERENCES

- [1] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.
- [2] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 09, 2020, pp. 13 693–13 696.
- [3] C. Yuan and S. S. Agaian, "A comprehensive review of binary neural network," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 949–13 013, 2023.
- [4] R. Wang, Y. Gong, X. Liu, G. Zhao, Z. Yang, B. Guo, Z. Zha, and P. Cheng, "Optimizing large language model training using fp4 quantization," *arXiv preprint arXiv:2501.17116*, 2025.
- [5] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, p. 107281, 2020.
- [6] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016.
- [7] H. Wang, S. Ma, L. Dong, S. Huang, H. Wang, L. Ma, F. Yang, R. Wang, Y. Wu, and F. Wei, "Bitnet: Scaling 1-bit transformers for large language models," *CoRR*, vol. abs/2310.11453, 2023.
- [8] P. Yin, J. Lyu, S. Zhang, S. J. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," in *International Conference on Learning Representations*, 2019.
- [9] J. Carrasquilla, M. Hibat-Allah, E. Inack, A. Makhzani, K. Neklyudov, G. W. Taylor, and G. Torlai, "Quantum hypernetworks: Training binary neural networks in quantum superposition," *arXiv preprint arXiv:2301.08292*, 2023.
- [10] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
- [11] M. Benedetti, B. Coyle, M. Fiorentini, M. Lubasch, and M. Rosenkranz, "Variational inference with a quantum computer," *Physical Review Applied*, vol. 16, no. 4, p. 044057, 2021.
- [12] J.-G. Liu and L. Wang, "Differentiable learning of quantum circuit born machines," *Physical Review A*, vol. 98, no. 6, p. 062324, 2018.
- [13] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [14] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *International Conference on Learning Representations*, 2017.
- [15] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, vol. 4, pp. 131–139, 1992.
- [16] A. Naim Ibrahim, "Scalable optimization of binary neural networks using variational quantum algorithms," Master's thesis, University of Guelph, 2024. [Online]. Available: <https://atrium.lib.uoguelph.ca/server/api/core/bitstreams/ff42e6c1-69b9-44bd-b0d2-1975854d5a0a/content>
- [17] G. E. Crooks, "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition," *arXiv preprint arXiv:1905.13311*, 2019.
- [18] J. Biamonte and V. Bergholm, "Tensor networks in a nutshell," *arXiv preprint arXiv:1708.00006*, 2017.
- [19] D. Molchanov, V. Kharitonov, A. Sobolev, and D. Vetrov, "Doubly semi-implicit variational inference," *arXiv preprint arXiv:1810.02789*, 2019.
- [20] J. Pomponi, S. Scardapane, and A. Uncini, "Bayesian neural networks with maximum mean discrepancy regularization," *Neurocomputing*, vol. 453, p. 428–437, Sep. 2021.
- [21] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012.
- [22] A. Müller, "Integral probability metrics and their generating classes of functions," *Advances in Applied Probability*, vol. 29, no. 2, p. 429–443, 1997.
- [23] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.
- [24] M. S. Rudolph, S. Lerch, S. Thanasilp, O. Kiss, O. Shaya, S. Vallecorsa, M. Grossi, and Z. Holmes, "Trainability barriers and opportunities in quantum generative modeling," *npj Quantum Information*, vol. 10, no. 1, p. 116, 2024.
- [25] S. Duffield, G. Matos, and M. Johanssen, "qujax: Simulating quantum circuits with JAX," *Journal of Open Source Software*, vol. 8, no. 89, p. 5504, Sep. 2023.
- [26] P. Glaser, M. Arbel, and A. Gretton, "Kale flow: A relaxed kl gradient flow for probabilities with disjoint support," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8018–8031, 2021.
- [27] T. L. Patti, J. Kossaifi, A. Anandkumar, and S. F. Yelin, "Variational quantum optimization with multibasis encodings," *Phys. Rev. Res.*, vol. 4, p. 033142, Aug 2022.