

Quantum Computing for Monitoring Methane Emissions

TNO 2026 R10798 – 24 August 2026

Quantum Computing for Monitoring Methane Emissions

Author(s)	Sven Müller, Esteban Aguilera, Simon Cramer, Shreshtha Sharma
Classification report	TNO Public
Number of pages	21 (excl. front and back cover)
Number of appendices	0

All rights reserved

No part of this publication may be reproduced and/or published by print, photoprint, microfilm or any other means without the previous written consent of TNO.

©2026 TNO

Contents

Contents	3
1 Introduction	4
2 Problem formulation	5
2.1 Current challenges	5
2.1.1 Updating methane emissions inventories.....	5
2.1.2 Plume detection	6
2.1.3 Synthetic data generation	7
2.2 Challenge selection	7
3 Solution	9
3.1 Dataset: MethaneS2CM	9
3.2 Training approach.....	9
3.3 Classical baseline.....	10
3.3.1 Original classical solution	10
3.3.1.1 Formulation	10
3.3.1.2 Limitations.....	10
3.3.2 Proposed classical solution.....	11
3.3.2.1 Formulation	11
3.3.2.2 Limitations.....	11
3.4 Quantum solutions.....	12
3.4.1 Quixer variant A.....	12
3.4.2 Quixer variant B.....	13
3.4.3 Limitations.....	16
4 Results.....	17
4.1 Metrics	17
4.2 Numerical results for the three variants	18
4.3 Interpretation of results.....	18
5 Conclusion	20
References.....	21

1 Introduction

Methane is one of the most important anthropogenic greenhouse gases, and timely detection of methane emissions is essential for climate monitoring, mitigation policy, and operational leak detection. In recent years, satellite-based Earth observation has become an increasingly relevant source of information for this purpose, as it enables large-scale monitoring of emission sources. At the same time, extracting actionable information from such data remains computationally challenging. For example, atmospheric inverse modeling and plume detection algorithms require methods that can handle high-dimensional inputs, temporal dependencies, and limited availability of labeled real-world examples. These characteristics make methane monitoring a useful application domain in which to assess whether emerging computational paradigms, including quantum computing, can offer practical added value.

Quantum computing is often considered as a promising route for accelerating selected computational primitives, such as linear algebra and optimization. However, in applied machine learning, the relevant question is not whether a theoretical speedup or improved learning capability exists, but whether it survives under realistic data-access, training, and output requirements. For that reason, this project evaluated where quantum approaches might plausibly contribute within the methane-monitoring pipeline. Three candidate challenges were considered in this pipeline. The first is the updating of methane emissions inventories through inverse modeling. The second is synthetic data generation. The third is plume detection from multi-temporal Sentinel-2 imagery. Following this initial assessment, a challenge was selected for quantum solution development and benchmarking. This report summarizes experimental results, insights, and recommendations for future work.

2 Problem formulation

Identifying where quantum computing can offer value requires the analysis of the computational structure of each candidate problem. This section describes three challenges arising in the context of methane emissions monitoring: updating methane emissions inventories through inverse modeling, generating synthetic training data for plume detection models, and detecting methane plumes in satellite imagery. For each challenge, we outline the relevant problem setting, current approaches, and the aspects where quantum methods could potentially contribute. Based on this assessment, one challenge is selected for further investigation in the remainder of the report.

2.1 Current challenges

2.1.1 Updating methane emissions inventories

Accurate methane (CH_4) emissions inventories are central to climate monitoring and policy: they quantify how much methane enters the atmosphere from human activities and natural processes, and they feed into the global assessments used to track progress against emissions targets. Maintaining up-to-date inventories, however, requires running computationally demanding atmospheric models, and the resulting computational bottlenecks motivate the search for more efficient algorithms.

CH_4 surface fluxes describe the rates at which methane is emitted or, to a small extent, absorbed at the Earth's surface. TNO AER (Air Quality and Emissions Research) estimates these fluxes using the TM5-4DVar inverse modeling system, which combines a chemistry-transport model (TM5-MP) with satellite and ground-based concentration observations. The approach is based on four-dimensional variational data assimilation (4D-Var): the model is run forward to produce simulated CH_4 concentrations, and those simulations are then compared with observations by minimizing a cost function that balances the fit to measurements against prior flux estimates, weighted by their respective error covariance matrices [1]. The primary output of this system is a set of monthly global CH_4 surface flux fields covering roughly the past 40 years.

The optimization relies on iterative gradient descent, so the cost-function gradient must be computed repeatedly. Two operations account for most of the computational cost: forward and adjoint runs of the TM5-MP transport model, and the inversion of large error covariance matrices. Both scale poorly with increasing spatial resolution or with longer assimilation windows, making them the limiting factors when higher-fidelity flux estimates are needed.

Quantum algorithms for solving linear systems have been proposed as a route to faster matrix inversion [2]. In principle these methods could speed up covariance inversion, but they face a fundamental readout obstacle in this context: any quantum speedup is conditioned on being able to query the result implicitly (through inner products or expectation values) rather than reading out every component of the solution vector. Computing the cost-function gradient, however, requires access to the full solution vector, component by component. The quantum speedup is therefore lost at the readout step, and no net advantage over classical methods is obtained. For this reason, the inverse-modeling problem is not taken forward in this study.

2.1.2 Plume detection

Detecting methane plumes from satellite imagery is a critical step toward quantifying point-source emissions at scale. Once a plume is detected, its observed concentration field can be translated into an emission rate, which can be used for monitoring industrial sources and informing regulatory efforts.

S&T is developing algorithms to detect methane plumes using Sentinel-2 and Sentinel-5 satellite data, following the deep learning approach of Rouet-Leduc et al. [3, 4]. Because deep learning models require large labeled training datasets, and real annotated plume images are scarce, S&T trains on synthetically generated data. The model performs well on held-out synthetic images, but its performance drops significantly when evaluated on real-world data [5].

Two directions were considered for applying quantum computing to this setting.

Enhancing the existing architecture. The first direction is to augment or replace components of the current segmentation model with quantum techniques:

- *Quantum-enhanced post-processing.* A graph-cut post-processing step can be formulated as a QUBO (quadratic unconstrained binary optimization) and solved with quantum optimization methods, incorporating additional spatial consistency information or Sentinel-5 data to refine segmentation masks [6, 7].
- *Quantum attention (Quixer).* Classical self-attention layers in a vision transformer can be replaced by a quantum attention mechanism based on linear combinations of unitaries (LCU) and the quantum singular value transformation (QSVT) [8]. This provides a different way of mixing information across inputs and is the approach ultimately pursued in this report.
- *Quantum knowledge distillation.* The S&T segmentation model can serve as a teacher network from which a smaller student network is trained by minimizing a loss that evaluates encoder representations using a quantum computer, potentially reducing the model size without sacrificing accuracy [9].
- *QuantumBoost.* A quantum boosting algorithm can train an ensemble of classical weak learners by computing approximate Bregman projections on a quantum computer; inference remains fully classical [10].

Training a quantum machine learning model end to end. The second direction is to build and train a quantum model directly on the real dataset. Generalization bounds for quantum machine learning suggest that the number of training samples required scales polynomially with the number of trainable parameters rather than with the (exponentially large) dimension of the Hilbert space [11]. This could be advantageous when labeled real-world data is limited, though empirical evidence on real, noisy datasets remains sparse. Candidate approaches include:

- *Quantum self-supervised learning*, which reduces dependence on large labeled datasets by leveraging unlabeled data during pre-training [12].
- *A variational quantum classifier*, following the approach used in an earlier TNO project with S&T on bathymetry estimation [13].

2.1.3 Synthetic data generation

Because large annotated datasets of real methane plumes are difficult to obtain, S&T trains its plume detection model on synthetic data. Synthetic plumes are created by embedding simulated plume absorption signatures into real Sentinel-2 background images. The simulation applies the Beer-Lambert law across a range of wind speeds and directions to produce physically plausible plumes.

The synthetic-to-real performance gap described in Section 2.1.2 may stem from multiple sources, one of which is a distributional mismatch between synthetic and real plumes: the simulated plumes may not reproduce the visual and spectral characteristics of plumes observed in actual satellite imagery. Improving the fidelity of synthetic data generation is therefore a natural complement to the model-level interventions considered above. Two directions were assessed.

Quantum speedup for the Beer-Lambert simulation. The Beer-Lambert computations are the most expensive part of the current synthetic data pipeline, and quantum algorithms could possibly accelerate them. However, a speedup alone would not change the output distribution of the generated images; faster simulation of the same physics model produces the same plumes, not more realistic ones. Since the goal is to reduce the distributional gap to real data rather than to reduce runtime, this direction was not pursued further.

Generative quantum models. Rather than simulating plumes from a physics model, one could train a generative model to sample realistic plume images. Generative quantum models have been shown to offer potential advantages over their classical counterparts in settings where the generative distribution is hard to sample classically but easy to train [14]. Specifically, a quantum advantage in this context can arise when (i) the model can be trained efficiently on a classical computer, but (ii) drawing new samples from the learned distribution is computationally hard to replicate classically. If real plume distributions satisfy these conditions, a generative quantum model could produce training data that more closely resembles real observations than the current physics-based approach.

2.2 Challenge selection

The three challenges were assessed to determine which was most suitable for a quantum computing approach. Plume detection was selected as the main focus, for the following reasons.

The inverse modeling challenge was set aside because the quantum speedup for matrix inversion breaks down at the readout step: the cost-function gradient requires all components of the solution vector, which eliminates the quantum advantage. The synthetic data generation challenge was also de-prioritized. Speeding up the Beer-Lambert simulation does not change the output distribution, and generative quantum models (while theoretically interesting) remain speculative in the absence of empirical evidence that real plume distributions satisfy the conditions required for a quantum advantage.

Plume detection, by contrast, offers a more concrete opportunity. The key property of the problem is its temporal structure: each sample consists of multiple Sentinel-2 observations of the same location at different points in time, and the discriminative information is encoded in how the scene changes across those observations. Quixer-style methods are well suited to this structure, since they allow the model to represent dependencies across the temporal context in a structured way. This makes the Quixer mechanism a plausible architectural

choice, which could also be implemented with existing attention mechanisms. Furthermore, the MethaneS2CM benchmark provides a concrete real-world dataset on which hypotheses can be tested and results reported.

The initial approach taken in this project was to fine-tune S&T's segmentation model to obtain a classical baseline, and then to complement it with quantum components. However, as had also been observed by S&T, these models (trained on S&T's synthetically generated data) did not generalize well to the real-data setting of the MethaneS2CM benchmark [5]. To reduce the possibility that this gap was caused by a mismatch between synthetic and real data sources, the model was also retrained directly on the real-data benchmark, where evaluation is performed on samples from unseen locations. Nonetheless, after several iterations, further improvements yielded diminishing returns. It was not clear which part of the segmentation model was the limiting factor, nor whether the available data contained sufficient predictive information for the task or whether that information was simply not being extracted effectively by the model. As a result, drawing conclusions about whether a quantum method would add value in the segmentation setting was not feasible, since the classical baseline was not yet sufficiently well understood.

For this reason, the study was redirected toward a simpler but closely related problem: tile-level binary classification of plume versus no-plume on the temporal MethaneS2CM dataset. Each sample in this dataset contains three Sentinel-2 observations of the same location at different points in time (the current image and two earlier reference images) making the setting temporal in nature. This formulation is easier to train and analyze, while remaining directly relevant to the overall plume-detection workflow. It allows us to establish whether temporal Sentinel-2 data contains enough predictive information for reliable classification, to obtain a well-understood classical baseline, and to evaluate whether Quixer-style methods add value in this controlled setting.

This classification task is not intended as a replacement for plume segmentation; rather, it serves as an intermediate step. A well-performing classifier is useful in its own right, but it may also function as a component in a later segmentation pipeline or as a diagnostic tool for identifying which aspects of the problem are learnable from the available data.

The question addressed in the remainder of this report is therefore the following:

Given three temporally ordered Sentinel-2 observations of the same location, can a Quixer-style classifier improve plume / no-plume prediction over a classical baseline under a shared encoder and evaluation setting?

3 Solution

3.1 Dataset: MethaneS2CM

The experiments in this report use the MethaneS2CM dataset [5], which was introduced as a real-world benchmark for methane-emission detection from Sentinel-2 multispectral data. As described in the accompanying paper [15], the dataset was constructed from Sentinel-2 imagery together with plume records and plume masks from the CarbonMapper platform. The aim of that work was to address several limitations of earlier approaches which were based mainly on simulated plumes. In particular, the paper shows that synthetic plumes overlaid on Sentinel-2 backgrounds may differ in distribution and visual characteristics from real-world data, and that using only a single reference image is often insufficient.

MethaneS2CM contains both L1C and L2A Sentinel-2 products and includes data from a large number of locations and time periods. Here, L1C refers to the top-of-atmosphere product, whereas L2A denotes the atmospherically corrected surface-reflectance product. For the experiments reported here, the L1C variant is used, which is also available through the Hugging Face release [5]. In the present classification setting, each sample is given as a 32×32 tile, together with a binary plume label and an associated plume mask. An important aspect of this dataset is that it is temporal, i.e., each sample contains multiple observations of the same location at different points in time. Each sample is represented by a leak-time observation together with two reference observations, approximately 90 days and 365 days earlier. This allows the model to use temporal changes in the scene rather than relying on a single image only.

3.2 Training approach

For the experiments reported in this work, nine Sentinel-2 bands are selected for each date, resulting in samples of dimension $9 \times 3 \times 32 \times 32$. The prediction target is the binary plume / no-plume label. All models investigated follow the structure depicted in Figure 3.1, where a CNN (Convolutional Neural Network) is used as common backbone for feature extraction, followed by a model-specific head that captures temporal interactions.

The training dataset (with 93,000 samples) and the test dataset (with 19,459 samples) correspond, respectively, to the periods before and after August 20, 2023. This temporal split is given by date, so the same location can be included in both datasets. The training dataset is further split as follows: 60,000 samples are used for fitting, 24,000 samples are reserved for validation and model selection, and 9,000 samples are reserved for post-refit calibration. The preprocessing is done as follows. First, the pixel values are scaled by a factor $1/10000$. Next, the band-wise mean and standard-deviation are computed based on the dataset for fitting. These same statistics are then reused for scaling the validation, calibration, and test datasets.

The validation step uses a grouped dataset with coordinate pairs as the grouping key, so that samples from the same location are kept together. Model selection is based on validation AU-ROC (Area Under the Receiver Operating Characteristic curve), with learning rate 4×10^{-3} and no early stopping. After model selection, the chosen configuration is refit on the combined fit and validation datasets. The calibration dataset is then used to tune the final decision threshold with the F1 objective, with a calibration fraction of 15% relative to the 60,000-sample fit

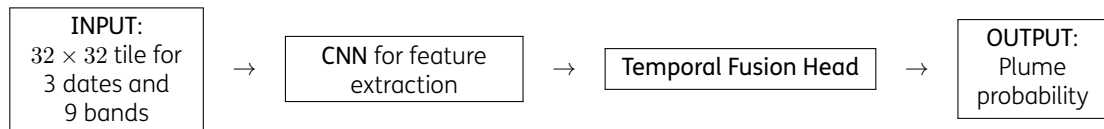


Figure 3.1: Architecture pattern used for all models in this report. A common CNN (Convolutional Neural Network) is used for feature extraction, followed by a model-specific head that captures temporal interactions.

dataset, which corresponds to the 9,000 calibration samples mentioned above. Threshold-dependent test metrics are therefore reported at a calibrated threshold, not at the common fixed threshold of 0.5.

3.3 Classical baseline

3.3.1 Original classical solution

3.3.1.1 Formulation

In the paper accompanying the MethaneS2CM dataset [15], the authors introduce a dedicated model, called MEECNet, for methane emission classification from Sentinel-2 data. The input consists of multiple observations of the same location at different times. The MEECNet model combines group convolutions with 3D convolutions. The role of the group convolutions is to process subsets of the spectral channels separately, so that the model can explicitly extract features from relevant band groups. The role of the 3D convolutions is then to aggregate information jointly across the temporal dimension and the learned feature maps.

The work in [15] evaluates two different tasks: binary image classification and image segmentation. For the binary classification task, the reported models are ResNet, ResNet3D, FCANet, ViT, and ViT-UNet, next to MEECNet itself. For the segmentation task, the reported baseline models are UNet, 3D U-Net, and Attention U-Net. This distinction is important in the present report, because our setting is a classification task rather than a segmentation task.

This is relevant here because it shows that the plume / no-plume classification task is not an artificial side problem introduced only for convenience in this report. It is also one of the main tasks used in the MethaneS2CM paper itself. At the same time, the paper already treats temporal information as an important part of the problem formulation. In that sense, the present work is aligned with the general direction of the work in [15], while asking a different question about how temporal information could be combined.

3.3.1.2 Limitations

For the purpose of the present study, the main limitation of [15] is not related to the reported performance, but rather to the complexity of the model for the purpose of classification only. This makes it less suitable as a controlled baseline, since our aim is to determine whether the Quixer mixing mechanism adds value. If one were to compare directly against a more specialized architecture such as MEECNet, it would become more difficult to determine whether any observed difference is caused by the Quixer component itself or by another component of the model. For this reason, the work in [15] is mainly used as motivation, rather than as a direct classical baseline.

3.3.2 Proposed classical solution

3.3.2.1 Formulation

The proposed classical baseline is designed to be simple and as comparable as possible to the Quixer variants studied in this report. The input consists of three Sentinel-2 observations of the same location at different points in time. This input is first split into three temporal inputs, each of which is given by:

$$x \in \mathbb{R}^{B \times C_d \times 32 \times 32}$$

where B is the batch size, C_d is the total number of multispectral bands per date for a 32×32 tile. A shared-weight per-date CNN encoder

$$f_\theta : \mathbb{R}^{C_d \times 32 \times 32} \rightarrow \mathbb{R}^{128}$$

is applied independently to each date:

$$z_t = f_\theta(x^{(t)}) \in \mathbb{R}^{128}, \quad t \in \{1, 2, 3\}.$$

The same parameter set θ is used for all three dates. The encoder f_θ consists of three convolutional blocks followed by global average pooling (GAP):

$$f_\theta = \text{GAP} \circ \text{Block}_3 \circ \text{Block}_2 \circ \text{Block}_1,$$

where each block has the form

$$\text{Block}_k = \text{MaxPool} \circ \text{ReLU} \circ \text{BatchNorm} \circ \text{Conv}.$$

More explicitly, the encoding evolves as

$$\mathbb{R}^{C_d \times 32 \times 32} \xrightarrow{\text{Block}_1} \mathbb{R}^{32 \times 16 \times 16} \xrightarrow{\text{Block}_2} \mathbb{R}^{64 \times 8 \times 8} \xrightarrow{\text{Block}_3} \mathbb{R}^{128 \times 4 \times 4} \xrightarrow{\text{GAP}} \mathbb{R}^{128}.$$

The three date embeddings are collected into

$$Z = [z_1, z_2, z_3] \in \mathbb{R}^{3 \times 128}.$$

The model then uses a multilayer perceptron (MLP) temporal fusion head. First, Z is flattened into

$$u = \text{vec}(Z) \in \mathbb{R}^{384}.$$

Then, the scalar logit is computed for binary plume / no-plume classification.

$$\ell = W_2 \phi(W_1 u + b_1) + b_2,$$

where

$$W_1 \in \mathbb{R}^{h \times 384}, \quad b_1 \in \mathbb{R}^h, \quad W_2 \in \mathbb{R}^{1 \times h}, \quad b_2 \in \mathbb{R},$$

h is the hidden dimension of the fusion head, and ϕ is the ReLU activation. The model is trained with binary cross-entropy.

The reason for choosing this baseline is that it preserves the temporal structure of the data while keeping the final fusion mechanism straightforward.

3.3.2.2 Limitations

The limitation of this classical baseline is that the interaction between the three temporal inputs only takes place at the very end of the model, through a standard MLP applied to the concatenated embeddings. In other words, the model can learn temporal correlations, but it does not impose any explicit structure on how information should be mixed.

3.4 Quantum solutions

3.4.1 Quixer variant A

The first quantum variant keeps the same shared CNN encoder but replaces the final MLP (in the classical baseline) by a Quixer [8] approach. Consequently, each date-specific input is initially encoded by the same shared-weight CNN yielding per-date embeddings

$$z_t = f_\theta(x^{(t)}) \in \mathbb{R}^{128}, \quad t \in \{1, 2, 3\}.$$

The part specific to this model is a quantum temporal fusion head. First, a learned map

$$W_E : \mathbb{R}^{128} \rightarrow \mathbb{R}^{4Lq}$$

produces

$$\Theta_t = \text{reshape}(W_E z_t + a_E) \in \mathbb{R}^{L \times 4 \times q},$$

where q determines the number of qubits in a quantum circuit and L is the number of circuit layers. The parameters Θ_t define the unitary

$$U_t(\cdot) = U(\cdot; \Theta_t) : \mathbb{C}^D \rightarrow \mathbb{C}^D.$$

with $D = 2^q$. Second, a learned map

$$W_b : \mathbb{R}^{128} \rightarrow \mathbb{R}^2$$

produces two real numbers

$$(\tilde{b}_t^{(R)}, \tilde{b}_t^{(I)}) = W_b z_t + a_b,$$

interpreted as the real and imaginary parts of a raw complex coefficient. Define

$$|\tilde{b}_t| = \sqrt{(\tilde{b}_t^{(R)})^2 + (\tilde{b}_t^{(I)})^2} + \varepsilon, \quad s = \sum_{t=1}^3 |\tilde{b}_t| + \varepsilon.$$

where $\varepsilon > 0$ denotes a small fixed constant introduced for numerical stability. The normalized complex mixing coefficient is then

$$b_t = \frac{\tilde{b}_t^{(R)}}{s} + i \frac{\tilde{b}_t^{(I)}}{s}.$$

Using the three operators U_1, U_2, U_3 and the three complex coefficients b_1, b_2, b_3 , the model defines the operator

$$M_Z : \mathbb{C}^D \rightarrow \mathbb{C}^D, \quad M_Z(v) = \sum_{t=1}^3 b_t U_t(v).$$

Thus, when M_Z acts on a reference state

$$v_0 = [1, 0, \dots, 0] \in \mathbb{C}^D$$

one obtains

$$M_Z(v_0) = \sum_{t=1}^3 b_t U_t(v_0).$$

That is, the same initial state v_0 is transformed by each date-specific operator, and the three resulting states are linearly combined.

The model subsequently applies a learned low-degree polynomial in the operator M_Z . Let $d \in \{2, 3\}$ denote the chosen polynomial degree, and let $c_0, c_1, \dots, c_d$ denote the normalized polynomial coefficients. The resulting state is

$$\psi = \sum_{k=0}^d c_k M_Z^k(v_0),$$

where

$$M_Z^0(v_0) = v_0, \quad M_Z^1(v_0) = M_Z(v_0), \quad M_Z^2(v_0) = M_Z(M_Z(v_0)),$$

and so on. After this stage, a second learned transformation

$$F : \mathbb{C}^D \rightarrow \mathbb{C}^D$$

is applied:

$$\psi' = F(\psi).$$

This transformation is again implemented by a quantum circuit, but its parameters are shared across samples and do not depend on the individual date embeddings. The resulting complex state ψ' is then mapped to a real-valued feature vector through a Pauli-style readout. Let

$$\bar{\psi} = \frac{\psi'}{\|\psi'\|_2 + \varepsilon}$$

denote the normalized state. For each qubit $j \in \{1, \dots, q\}$, the readout computes expectation-style quantities

$$\langle X_j \rangle, \quad \langle Y_j \rangle, \quad \langle Z_j \rangle.$$

These are concatenated into a vector

$$r \in \mathbb{R}^{3q}.$$

Finally, a standard MLP readout

$$g : \mathbb{R}^{3q} \rightarrow \mathbb{R}$$

maps the measurement vector to the scalar logit to be used for binary plume / no-plume classification.

To summarize, the model first extracts three per-date embeddings using the same CNN as the classical baseline, then uses those embeddings to define a sample-specific mixed operator M_Z , applies a learned polynomial in that operator to a fixed reference state, transforms the resulting state once more, measures it through Pauli-style observables, and maps the resulting real feature vector to the final logit.

3.4.2 Quixer variant B

The second Quixer variant developed in this project is experimental and does not follow the circuit model that would allow it to run a quantum computer. As such, it can be considered to be a quantum-inspired model only. Again, this model keeps the same shared CNN encoder and replaces the final MLP by a Quixer approach. As before, the initial embeddings are given by

$$z_t = f_\theta(x^{(t)}) \in \mathbb{R}^{128}, \quad t \in \{1, 2, 3\},$$

which are collected into

$$Z = [z_1, z_2, z_3].$$

The model-specific temporal fusion head maintains a real-valued state

$$s_k \in \mathbb{R}^F$$

at iteration k , where

$$D = 2^q, \quad F = 2D = 2^{q+1},$$

and q is the qubit count. The factor 2 is used because the state is intended to represent the real and imaginary parts of a complex vector in $\mathbb{C}^D$. The state is initialized at the fixed reference vector

$$s_0 = [1, 0, \dots, 0] \in \mathbb{R}^F.$$

At each iteration k , the model constructs a joint update from the three date embeddings. First, it computes global mixing logits

$$a(Z) \in \mathbb{R}^3$$

via a learned MLP. These logits are converted into mixing weights

$$\alpha(Z) = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3.$$

Depending on the configuration, either a softmax normalization is used,

$$\alpha_t = \frac{e^{a_t}}{\sum_{j=1}^3 e^{a_j}},$$

or an L_1 normalization is used,

$$\alpha_t = \frac{a_t}{\sum_{j=1}^3 |a_j| + \varepsilon},$$

where $\varepsilon > 0$ is a small fixed constant used for numerical stability. Next, each date embedding z_t is mapped to a scalar gate

$$g_t = \sigma(w_g^\top z_t + b_g) \in (0, 1),$$

where σ is the logistic sigmoid, $w_g \in \mathbb{R}^{128}$, and $b_g \in \mathbb{R}$. This gate controls the strength of the update proposed by date t .

The core model-specific quantity is the date-specific update proposal

$$u_t(s_k, z_t) \in \mathbb{R}^F,$$

which specifies how date t proposes to modify the current state s_k . The exact form of u_t depends on the chosen operator family.

Low-rank operator family. If a low-rank operator family is used, each date embedding z_t is mapped to two matrices

$$U_t \in \mathbb{R}^{F \times r}, \quad V_t \in \mathbb{R}^{F \times r},$$

where r is the configured low-rank rank. These matrices are produced by learned affine maps applied to z_t . The proposal is then

$$u_t(s_k, z_t) = U_t(V_t^\top s_k).$$

Thus, each date defines a sample-specific low-rank linear operator acting on the current state.

Householder operator family. If a Householder-type operator family is used, each date embedding z_t is mapped to a target vector

$$v_t = \tanh(W_h z_t + b_h) \in \mathbb{R}^F,$$

which is normalized as

$$\hat{v}_t = \frac{v_t}{\|v_t\|_2 + \varepsilon}.$$

Let s_0 denote the reference state. Define

$$d_t = s_0 - \hat{v}_t.$$

From this, a reflection direction is formed by normalizing d_t . This yields a reflection operator R_t acting on the current state s_k . The proposal is then

$$u_t(s_k, z_t) = R_t(s_k) - s_k.$$

Hence, each date proposes a reflection-induced correction of the current state.

Dense operator family. If a dense operator family is used, the model stores three learned matrices

$$A_t \in \mathbb{R}^{F \times F}, \quad t \in \{1, 2, 3\},$$

shared across all samples, and defines

$$u_t(s_k) = A_t s_k.$$

In this case the operator itself does not depend directly on z_t , but the final contribution of date t still depends on z_t through the gate g_t and the global mixing weight α_t .

Once the three date-specific proposals are available, they are combined into a single joint update

$$\Delta(s_k; Z) = \sum_{t=1}^3 \alpha_t(Z) g_t(z_t) u_t(s_k, z_t).$$

This equation is the core of the temporal fusion head: each date proposes an update to the current state, each proposal is amplitude-controlled by its gate, and the gated proposals are mixed using weights that depend on the full date triple.

Let K denote the configured number of update steps. For each iteration $k = 0, \dots, K-1$, the model also uses a learned step size

$$\eta_k = \sigma(\gamma_k) \in (0, 1),$$

where γ_k is a learned scalar. The state is updated according to the rule

$$s_{k+1} = \mathcal{N}(s_k + \eta_k \Delta(s_k; Z)).$$

Here $\mathcal{N}$ is a chosen normalization operator. After K iterations, the model obtains a final state

$$s_K \in \mathbb{R}^F.$$

The final readout is performed in one of two modes.

Direct MLP readout. In direct readout mode, the final state is mapped directly to a scalar logit:

$$\ell = g_{\text{MLP}}(s_K),$$

where

$$g_{\text{MLP}} : \mathbb{R}^F \rightarrow \mathbb{R}$$

is a learned two-layer MLP.

Pauli-style readout. In Pauli-style readout mode, the final real state is interpreted as a complex vector. Write

$$s_K = [r_K; m_K], \quad r_K, m_K \in \mathbb{R}^D,$$

and define

$$\psi_K = r_K + i m_K \in \mathbb{C}^D.$$

A Pauli-style measurement map

$$\Phi : \mathbb{C}^D \rightarrow \mathbb{R}^{3q}$$

then computes X -, Y -, and Z -type expectation features for each of the q qubits. A final MLP

$$g_{\text{Pauli}} : \mathbb{R}^{3q} \rightarrow \mathbb{R}$$

maps those features to the scalar logit:

$$\ell = g_{\text{Pauli}}(\Phi(\psi_K)).$$

To summarize, the model first extracts three per-date embeddings using the same shared CNN as the classical baseline, then iteratively refines a state by combining three gated date-specific update proposals, normalizes the state after each residual step, and finally maps the resulting state to the output either directly or through a Pauli-style measurement layer. The iterative update rule gives the model a small residual dynamical system. This can increase expressiveness relative to a single fusion layer while keeping the number of update steps fixed and small.

3.4.3 Limitations

Compared with the classical baseline, these variants introduce additional design choices, such as the operator family, the normalization steps, and the readout type. In this study, these choices are fixed to one selected configuration, but the resulting performance should still be interpreted together with this additional inductive bias and modeling flexibility.

4 Results

Note that the results presented in this section are based on small qubit counts, which makes it possible to implement quantum variant A without the need of a quantum computer; quantum operations are performed through standard packages for linear algebra. As a result, improved performance in this setting would show that the mixing mechanism is useful, but it would not by itself imply a quantum advantage in the computational sense.

4.1 Metrics

For binary plume / no-plume classification, a model first produces a probability for each sample. To obtain a hard prediction, one chooses a threshold τ . Samples with predicted probability above τ are classified as plume, while all others are classified as no plume. This threshold choice is also referred to as the operating point of the classifier. Some metrics depend on this threshold choice, while others do not. In particular, AUROC and AUPRC (see definitions below) summarize how well the model ranks positive samples above negative ones across a range of thresholds, rather than at one fixed threshold. The Brier score instead measures probability calibration, that is, whether predicted probabilities numerically agree with the observed frequencies of positive cases. Specifically,

- AUROC is the area under the receiver operating characteristic curve, which plots the true-positive rate against the false-positive rate across thresholds. AUROC represents the probability that a model will rank a randomly chosen positive instance higher than a randomly chosen negative one.
- AUPRC is the area under the precision-recall curve, which is particularly informative when false alarms are costly or the class balance is uneven.
- Accuracy, precision, recall, and F1 are reported at the final calibrated threshold τ^* , selected on the grouped post-refit calibration split using the F1 objective.
- Recall@1%FPR reports the recall obtained at the threshold for which the false-positive rate is 1%.
- Precision@90%Recall reports the precision obtained at the threshold for which the recall is 90%.
- The Brier score is the mean squared difference between predicted probabilities and binary labels, and therefore summarizes calibration quality.

The evaluation approach used in this report includes a final calibration step, so the threshold-dependent metrics and the threshold-independent metrics should be interpreted in different ways. AUROC and AUPRC describe how well the model orders the samples across thresholds. The Brier score describes how reliable the predicted probabilities are. Accuracy, precision, recall, and F1 describe performance at the final selected threshold τ^* . Recall@1%FPR and Precision@90%Recall describe two additional constrained operating regimes and are therefore also threshold-based, but not tied specifically to τ^* .

Solution	τ^*	AUROC	AUPRC	Accuracy@ τ^*	Precision@ τ^*	Recall@ τ^*	F1@ τ^*
Classical baseline	0.280	0.8288	0.8338	0.7385	0.6981	0.8471	0.7654
Quixer variant A	0.226	0.8339	0.8285	0.7370	0.6871	0.8776	0.7707
Quixer variant B	0.229	0.8413	0.8355	0.7495	0.7091	0.8524	0.7741

Table 4.1: Representative classification results. Threshold-dependent metrics are reported at the calibrated post-refit threshold τ^* .

Solution	Brier	Recall@1%FPR	Precision@90%Recall
Classical baseline	0.1797	0.1718	0.6524
Quixer variant A	0.1744	0.1450	0.6694
Quixer variant B	0.1765	0.1504	0.6721

Table 4.2: Additional operating-point metrics for the same representative runs. These emphasize calibration and behavior in stricter decision regimes.

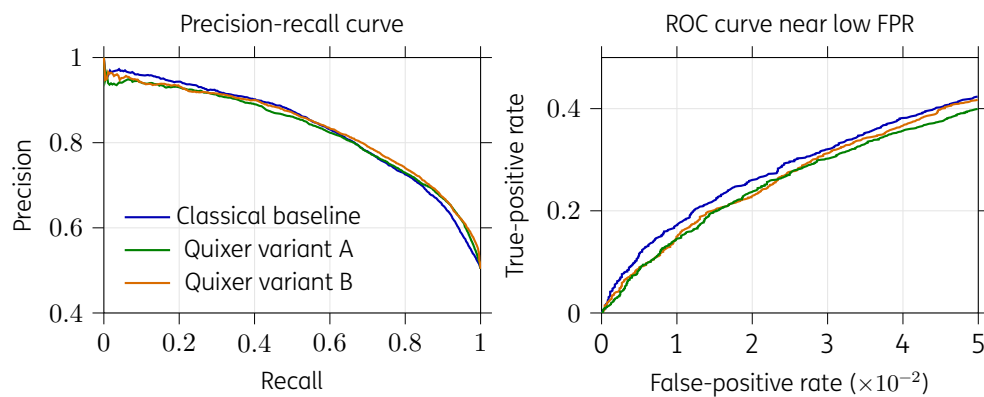


Figure 4.1: Precision-recall curves and low-FPR ROC curves for the three representative runs reported in Table 4.1. The ROC panel is restricted to the 0–5% false-positive range because this is the part most relevant to the low-FPR discussion in the text.

4.2 Numerical results for the three variants

Tables 4.1 and 4.2 summarize representative runs used for the three reported solutions with random seed 200 under the grouped validation and post-refit calibration approach.

For the classical baseline and Quixer variant B (both trained with 40 epochs), additional runs with random seeds 201 and 202 are available and are shown in Figure 4.2 for a consistency check. For the Quixer variant A, the selected 60-epoch configuration is represented only by the run with random seed 200.

Figures 4.1 and 4.2 provide two complementary views on the same reported runs. Figure 4.1 compares ranking behavior across thresholds and in the low-false-positive regime for the representative runs. Figure 4.2 summarizes the available multi-seed results.

4.3 Interpretation of results

The reported comparison can be read at three different levels: ranking quality across thresholds, performance at the calibrated operating point, and performance in more constrained operating regimes. Taken together, these layers support four main observations.

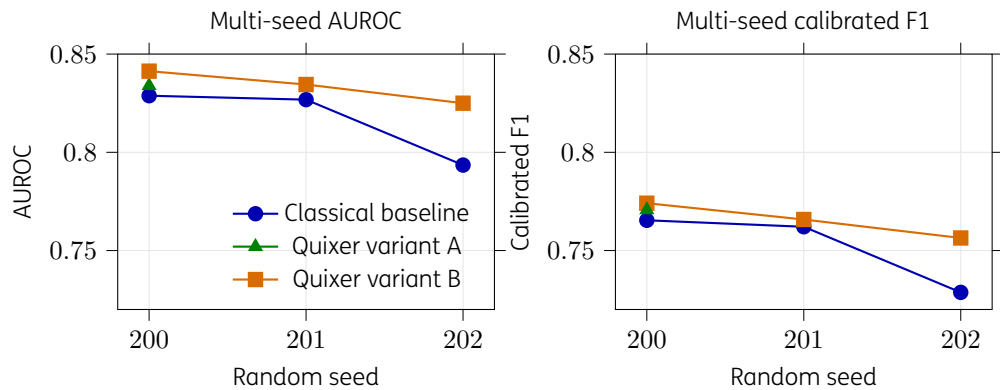


Figure 4.2: Multi-seed results. The classical baseline and Quixer variant B each have runs for random seeds 200, 201, and 202. For the final selected Quixer variant A configuration, only the run with random seed 200 was available.

First, the Quixer variant B gives the best overall result among the reported runs, where “overall” is understood as combining strong ranking quality with strong performance at the calibrated operating point. It reaches the highest AUROC (0.8413) and the highest AUPRC (0.8355), while also obtaining the strongest calibrated F1 score (0.7741). However, the precision-recall curves in Figure 4.1 show that this advantage is not uniform across the entire curve. In the high-precision, low-recall regime, the classical baseline sits above the other variants, whereas the Quixer variant B is stronger over a broader part of the curve and therefore achieves the best overall AUPRC. At the same time, the margins are modest, so this result should not be overstated.

Second, the three models differ in how they behave at the calibrated threshold τ^* , which is selected for all models by the same grouped post-refit calibration procedure using the F1 objective. The classical baseline reaches calibrated recall 0.8471 with precision 0.6981. The Quixer variant B slightly improves precision, F1, and recall level. The Quixer variant A reaches the highest calibrated recall of the three (0.8776), but does so with the lowest calibrated precision (0.6871). In this sense, Quixer variant A is the most recall-oriented solution, whereas Quixer variant B gives the most balanced operating point.

Third, the constrained operating-regime metrics show a more mixed picture. The Quixer variant A gives the lowest Brier score (0.1744), indicating slightly better overall probabilistic accuracy among the reported runs. However, the best Recall@1%FPR is obtained by the classical baseline (0.1718). This indicates that improvements in global ranking quality, as reflected by AUROC and AUPRC, do not necessarily improve performance in the extreme low-false-positive regime, which may be the most operationally relevant setting. This is also visible in the low-FPR ROC panel of Figure 4.1, where the classical baseline is slightly ahead of the other two variants near the 1% false-positive region.

Fourth, the available multi-seed evidence favors the Quixer variant B solution over the classical baseline. Across the three available seeds, the classical model reaches mean AUROC 0.8164 and mean calibrated F1 0.7521, whereas the Quixer variant B reaches mean AUROC 0.8336 and mean calibrated F1 0.7654. More importantly, Figure 4.2 shows that the Quixer variant B results vary less across seeds than the classical ones. In that sense, the Quixer variant B is not only stronger on average, but also more stable (although a very small number of seeds have been considered). For Quixer variant A, the final 60-epoch result is currently available only for the run with random seed 200. As a result, comparisons involving the Quixer variant A configuration are less robust than the comparison between the classical baseline and the Quixer variant B.

5 Conclusion

This report examined where quantum methods might contribute within the methane monitoring pipeline. Three candidate challenges were considered: updating methane emissions inventories through inverse modeling, generating synthetic training data, and improving plume detection from satellite imagery. Of these, plume detection was selected as the most concrete and testable setting. The study therefore focused on a controlled binary classification task on the temporal MethaneS2CM benchmark, asking whether a quantum temporal fusion mechanism can improve plume / no-plume prediction relative to a straightforward classical baseline under a shared encoder and evaluation procedure.

The reported experiments confirm previous results [5] that show that temporal Sentinel-2 observations contain usable predictive information for methane plume classification in the evaluated setting. All three investigated models were able to extract signal from the three-date input, and the quantum variants were competitive with the classical baseline. At the same time, the observed margins remain modest and depend on the evaluation regime. The results should therefore be interpreted as evidence that the temporal fusion mechanism matters, rather than as evidence of superiority of one modeling paradigm across all deployment scenarios.

The main implication of these findings is not that quantum computing has demonstrated a practical computational advantage for methane monitoring. The evaluated quantum methods were carried out without the use of a quantum computer, and improved performance in this setting would therefore indicate that a quantum-motivated inductive bias can be useful for temporal remote-sensing classification, rather than that a quantum computer solves the task more efficiently. The study also confirms the value of simplifying the problem to tile-level classification as an intermediate step: this formulation makes it possible to assess whether the data contains learnable temporal signal and whether alternative fusion heads can exploit it in a controlled comparison.

Several limitations remain. The comparison is limited in scale, the available multi-seed evidence is still narrow, and additional ablations are needed to separate the effect of the temporal mixing principle from the effect of added modeling flexibility. In particular, the quantum heads introduce additional design choices, and the present results should be interpreted together with those choices rather than as a pure test of one isolated mechanism. Moreover, the present study addresses tile-level classification rather than plume segmentation, whereas plume localization remains a key operationally relevant objective. Future work should therefore extend the comparison to a broader set of random seeds, include stronger and carefully matched temporal classical baselines, perform targeted ablation studies on the fusion-head design, and test whether the observed benefits transfer to segmentation and to operating regimes that reflect practical deployment constraints. Taken together, the results support a narrow but relevant conclusion: quantum model design can be a useful source of new temporal fusion mechanisms for methane-plume detection, but the case for broader operational value still requires more extensive empirical validation.

References

- [1] A. Segers and R. Nanni. *CAMS2_55_CEA – Provision of global inversion-optimised greenhouse gas fluxes and concentrations: Contribution to documentation of products and services as provided within the scope of this contract – 2023 – Part CH4*. Version 1; accessed 2026-03-03. TNO, 2023.
- [2] M. Benedetti, A. Rosmanis, and M. Rosenkranz. “A probabilistic quantum algorithm for Lyapunov equations and matrix inversion”. In: *Physical Review Applied* (Mar. 2026). DOI: 10.1103/MPRK-9VX3.
- [3] B. Rouet-Leduc, T. Kerdreux, A. Tuel, and C. Hulbert. *Autonomous Detection of Methane Emissions in Multispectral Satellite Data Using Deep Learning*. Aug. 2023. DOI: 10.48550/arXiv.2308.11003.
- [4] B. Rouet-Leduc and C. Hulbert. “Automatic detection of methane emissions in multispectral satellite imagery using a vision transformer”. In: *Nature Communications* 15.1 (2024), p. 3801. ISSN: 2041-1723. DOI: 10.1038/s41467-024-47754-y.
- [5] H. Liu, J. Y. Leung, and D. Niu. *H1deaki/MethaneS2CM*. Dataset card includes a recommended citation to the accompanying KDD 2025 paper; accessed 2026-03-03. Hugging Face, 2025.
- [6] Y. Y. Boykov and M.-P. Jolly. “Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images”. In: *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001)*. Vol. 1. IEEE, 2001, pp. 105–112. DOI: 10.1109/ICCV.2001.937505.
- [7] L. Domingo Colomer and M. Chehimi. *Quantum-enhanced unsupervised image segmentation for medical images analysis*. Nov. 2024. DOI: 10.48550/arXiv.2411.15086.
- [8] N. Khatri, G. Matos, L. Coopmans, and S. Clark. *Quixer: A Quantum Transformer Model*. June 2024. DOI: 10.48550/arXiv.2406.04305.
- [9] C.-Y. Liu et al. *Quantum Relational Knowledge Distillation*. Aug. 2025. DOI: 10.48550/arXiv.2508.13054.
- [10] A. Abbas, Y. Chen, T. Nguyen, and R. de Wolf. *QuantumBoost: A lazy, yet fast, quantum algorithm for learning with weak hypotheses*. Oct. 2025. DOI: 10.48550/arXiv.2510.05089.
- [11] M. C. Caro et al. “Generalization in quantum machine learning from few training data”. In: *Nature Communications* 13 (2022), p. 4919. DOI: 10.1038/s41467-022-32550-3.
- [12] B. Jaderberg et al. *Quantum Self-Supervised Learning*. Mar. 2021. DOI: 10.48550/arXiv.2103.14653.
- [13] E. Aguilera, R. Wezeman, J. Senior-Williams, and E. van Mulligen. *TNO Quantum Variational Classifier: Powering ML workflows with quantum*. PennyLane Blog. Accessed 2026-03-03. Aug. 2023.
- [14] H.-Y. Huang et al. *Generative quantum advantage for classical and quantum problems*. Sept. 2025. DOI: 10.48550/arXiv.2509.09033.
- [15] H. Liu, J. Y. Leung, and D. Niu. “MethaneS2CM: A Dataset for Multispectral Deep Methane Emission Detection”. In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025, pp. 5640–5651.

ICT, Strategy & Policy

Anna van Buerenplein 1
2595 DA Den Haag
www.tno.nl