

METHOD STUDY

The Sensitivity Map of Quantum Classifiers

One dial decides, and the model can learn it. Which settings move the score of a quantum classifier on real data, and which do not. Eight settings, 360 trainings, two public credit datasets, one internal dataset and a synthetic control, all on qubit-lab.ch's Rapid Quantum Prototype for machine learning (QML RQP Pro): the product used as an evidence-driven experimentation platform.

November 2026

qubit-lab.ch

qubit-lab.ch
contact@qubit-lab.ch

© 2026 qubit-lab.ch · All rights reserved · qubit-lab.ch/legal

Contents

For decision makers: the study on one page . . . 2
Five terms, one line each . . . 2
1. The study at a glance . . . 3
2. Finding 1: one dial decides . . . 4
3. Finding 2: loading repeats are the same dial . . . 5
4. Finding 3: the method sees a signal where there is one . . . 6
5. Finding 4: the model can learn the dial . . . 6
6. Finding 5: the cheapest circuit is the right one . . . 8
7. Finding 6: what you can leave alone . . . 9
8. Finding 7: the random start is a setting . . . 10
9. Try it on your table . . . 11
10. Where the classifier stands against standard models . . . 11
11. Limits . . . 11
Appendix A. The datasets in detail . . . 13
German Credit . . . 13
Taiwan credit default . . . 13
Teacher control . . . 14
The internal dataset . . . 14
Appendix B. Settings, recipe and definitions . . . 14
Appendix C. The registered hypotheses . . . 15
Appendix D. The decision table and the product . . . 16
Appendix E. Record . . . 16
Disclaimer . . . 17

© qubit-lab.ch

For decision makers: the study on one page

The question. A quantum classifier is a machine-learning model that runs on a quantum computer, or on a simulator of one. Before anyone can try it on a bank's data, about eight technical settings have to be chosen. Nobody had measured which of them matter on real financial data, so every trial began with guesswork, at one training run per guess.

What we found, in 360 training runs on four datasets.

1. **One setting decides.** How strongly the data is scaled when it is loaded into the quantum circuit moves the result by 13 to 16 points on a 100-point scale on public credit data. Six of the eight settings can stay at their factory defaults.
2. **The model can find that setting itself.** A new option of the November 2026 release lets the model learn the scale during training. One run then lands within one point of the best result that a search over 27 configurations had found by hand.
3. **The simplest circuit is enough.** The smallest standard circuit was never beaten by a larger one by more than the normal variation between two runs. Smaller circuits are also the cheaper ones to run on quantum hardware.

How it was measured. Every training is a run of the QML RQP Pro, and every setting is a setting of that product.

What this means for a pilot. Checking a quantum classifier on your own data takes four to seven runs, each between five minutes and an hour: an afternoon, not a project. The effort does not go into circuit design.

Where the technology stands. On the two public credit datasets the quantum classifier is on par with a standard statistical model on one and below it on the other. On a control dataset that was built to contain a pattern of the quantum kind, it reaches 79 points where the standard models stay between 33 and 42. This study does not claim an advantage on real financial data. The largest factor is the data: the same classifier reaches 79 points on the control and 62 to 66 on the credit tables. The second is the scale. The study makes it cheap to find out what your own data holds.

What to do. If quantum machine learning is on your innovation agenda, run the short check of section 9 with the same tool on one of your own tables before commissioning anything larger.

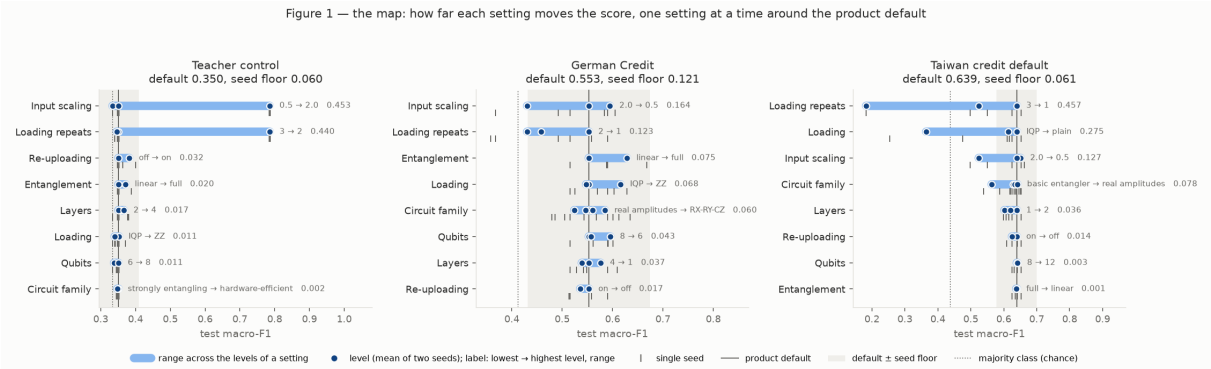


Figure 1. How far each of the eight settings moves the result on the three published datasets. The reference line is the normal variation between two runs of the same configuration. A setting whose bar stays under the line needs no tuning.

Five terms, one line each

Term	Meaning
Qubit	The unit of a quantum computer. This study uses 6 to 12 of them, simulated exactly.
Loading method	How a row of data is written into the qubits: each feature turns one qubit by an angle.
Scale	How far a feature's usual range turns its qubit. The one setting that decides here.
Score	The quality of the predictions on rows the model has not seen, from 0 to 100 points. A model that always predicts the larger class scores 33 to 44 points, depending on the dataset.
Random start	Training begins from random values, so two runs of the same configuration differ. The usual size of that difference is the **noise level** (6 to 12 points here). An effect smaller than the noise level is not an effect.

1. The study at a glance

The question in practice. Training a quantum classifier means choosing how the data is loaded into the qubits (three ways), how many times it is loaded, whether it is loaded again between the processing layers, how strongly the inputs are scaled, how the qubits are tied together, how many processing layers there are, how many qubits, and which circuit family does the processing. A data scientist who meets these settings for the first time asks: on my kind of data, which of these is worth turning, and which can I leave alone? This study answers with a map.

What was run.

- **Instrument.** QML RQP Pro, the variational classifier, as deployed on 6 November 2026 and pinned for the whole study. Every number is the product's own test score of a product run.
- **The same training recipe for every run**, the one of the product's published sample workbook (details in Appendix B).
- **Simulation without noise.** No device was used. Which configuration keeps its score on hardware is the question of the next study.
- **Two random starts per configuration**, so that every comparison can be read against the noise level.

Stage	What	Trainings
1	Every setting one at a time around the default, plus all 27 combinations of loading, loading repeats and scale ("the corner")	286
2	The levers the map suggested: trainable scaling, a richer readout, more training rows, an ensemble, and a third random start on German Credit	56
3	The three circuit families of the product that stage 1 had not covered	18
	Total	**360**

Of the 360 trainings, 275 are on the three published datasets and 85 on one internal dataset. No training in the ledger failed or was excluded. One batch is not in the count: the first 70 trainings on the control were discarded and repeated after a preparation error was found (Appendix A).

The datasets. Appendix A describes each one in full.

	German Credit	Taiwan credit default	Teacher control	Internal dataset
What it is	1,000 loan applications, public	30,000 credit-card customers, public	1,000 synthetic rows with a known quantum pattern	a large real fraud table
Role	the benchmark of the literature; small	the precise instrument: 6,000 test rows	shows what the method can see	the client-like case; numbers not shown
Chance score	0.412	0.438	0.333	not shown
Logistic regression	0.668	0.648	0.333	not shown
Noise level	0.121	0.061	0.060	not shown

Reading the numbers. The text gives scores in points from 0 to 100; the tables print the same values as decimals, so 0.616 in a table is 61.6 points. A configuration's score is the mean of its two random starts. The noise level of a dataset is the difference between two starts that four out of five configurations stay under; it was defined before the first run (Appendix B). The "range" of a setting is the highest minus the lowest score across its levels, with the other seven settings at their defaults.

2. Finding 1: one dial decides

Range of each setting	Teacher control	German Credit	Taiwan credit default
Input scaling	**0.453**	**0.164**	**0.127**
Loading repeats	**0.440**	**0.123**	**0.457**
Loading	0.011	0.068	**0.275**
Circuit family	0.002	0.060	**0.078**
Entanglement	0.020	0.075	0.001
Layers	0.017	0.037	0.036
Qubits	0.011	0.043	0.003
Re-uploading	0.032	0.017	0.014
Noise level	0.060	0.121	0.061

Bold: the range is above the dataset's noise level. This is Figure 1 in numbers.

The input scaling and the loading repeats clear the noise level on every dataset, the internal one included. Four settings never do: re-uploading, entanglement, layers and qubits. The two remaining bold entries on Taiwan, loading and circuit family, are each a single level that fails; sections 3 and 7 explain them.

What the dial does. The loading turns each qubit by an angle that is proportional to the feature value. The input scaling sets how far: at the default of 1.0, a value at the outer edge of a feature's usual range turns its qubit by half a turn; at 0.5 by a quarter turn; at 2.0 by a full turn. A full turn brings the qubit back to where it started, so at scale 2.0 a large feature value and a small one end in the same state. The feature's range is wrapped around, and the classifier sees an oscillating pattern where the data has a simple trend.

Why real data wants a gentle scale. In credit data the label mostly rises or falls with a feature: more arrears, more defaults. A gentle scale keeps that trend visible. On both public datasets the score is on a plateau at scale 0.5 and 1.0 and falls off at 2.0:

Plain loading, one repeat	Scale 0.5	Scale 1.0 (default)	Scale 2.0
German Credit	0.595	0.553	0.430
Taiwan credit default	0.651	0.639	0.524
Teacher control	0.333	0.350	0.786

The product default of 1.0 sits on the edge of the plateau on both credit datasets: usable, with a small gain left at 0.5. The control wants the opposite, a wide scale, and section 4 comes back to it.

What happens beyond the edge. When the range is wrapped too far, training stalls. The pattern oscillates faster than the circuit can follow, the output becomes a constant, and the model ends up calling everything positive. On Taiwan with three loading repeats both random starts end at 18.2 points, within 0.1 point of the score of a model that calls every customer a defaulter. This is the one expensive mistake in the whole map, and it is easy to avoid.

What this means for you. Before judging a quantum classifier on your data, check the scale. A poor result at the wrong scale says nothing about the method.

3. Finding 2: loading repeats are the same dial

Repeating the plain loading applies the same rotation again, so two repeats turn each qubit twice as far. Loading repeats and input scaling multiply. The corner measured this directly:

Plain loading	Teacher control	German Credit	Taiwan credit default
1 repeat, scale 2.0	0.786	0.430	0.524
2 repeats, scale 1.0	0.786	0.430	0.524
1 repeat, scale 1.0	0.350	0.553	0.639
2 repeats, scale 0.5	0.350	0.553	0.639

The pairs agree to the last digit because they are the same circuit. For the two richer loadings (ZZ and IQP) the relation is approximate, because their extra gates carry products of two features and grow faster than the single rotations.

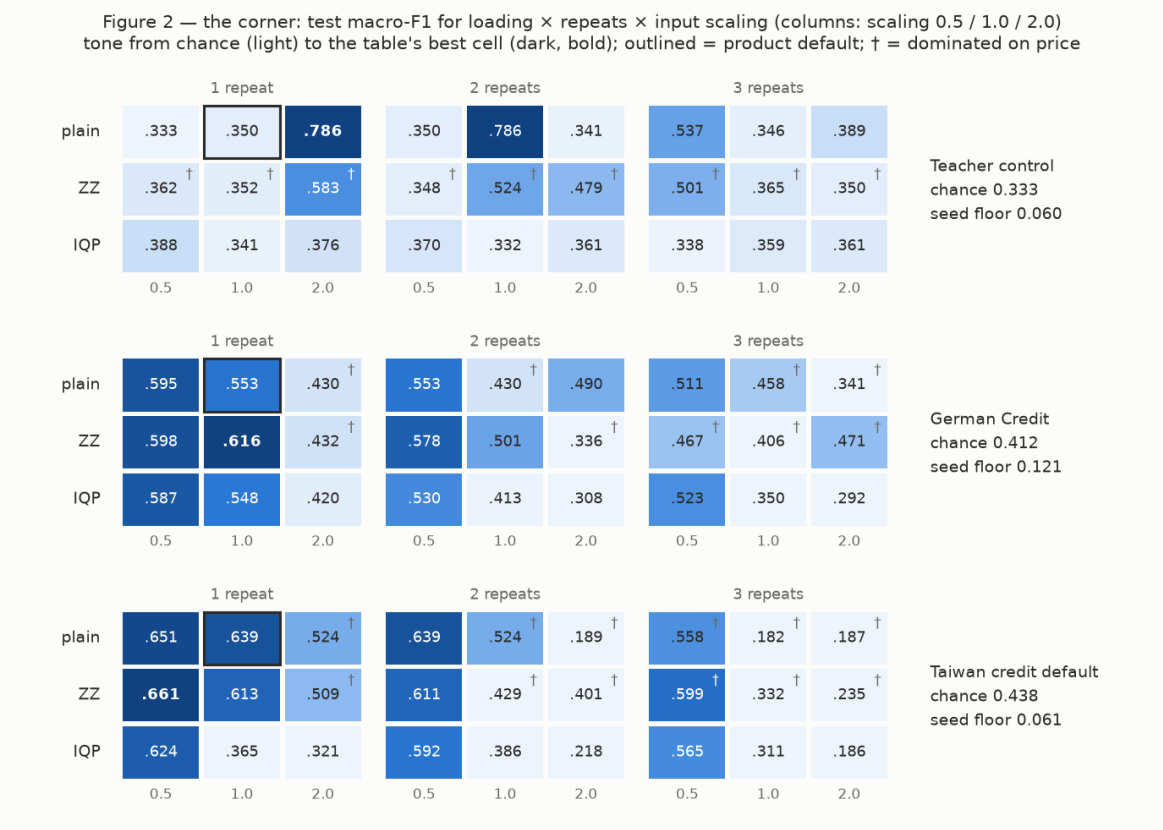


Figure 2. All 27 combinations of loading, repeats and scale per dataset.

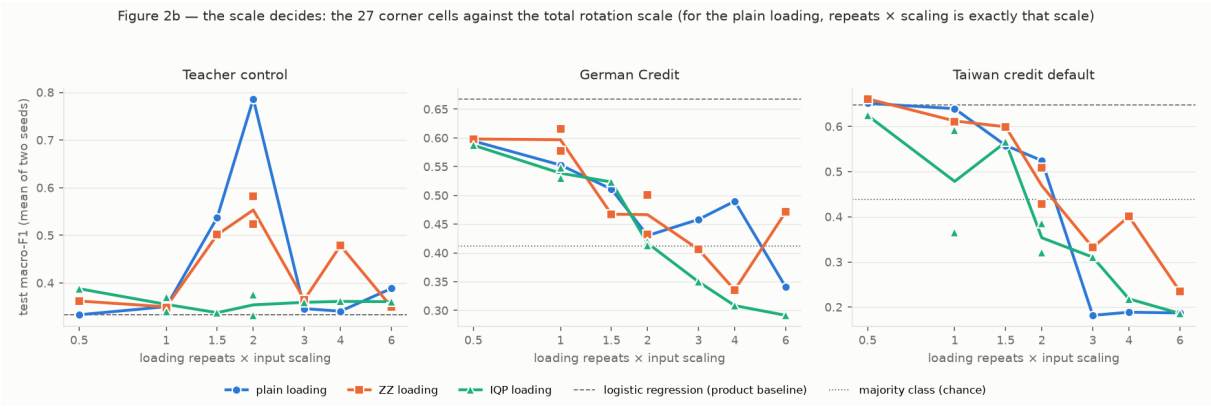


Figure 2b. The same 27 cells as curves over the total scale, the product of loading repeats and input scaling.

Across the 27 corner cells, scale and repeats together account for 72 % of the differences between cells on German Credit and 77 % on Taiwan. The kind of loading accounts for 6 % and 5 %.

Share of the differences between corner cells	Teacher control	German Credit	Taiwan credit default
Scaling	3 %	49 %	55 %
Repeats	2 %	23 %	22 %
Loading	13 %	6 %	5 %
Repeats × scaling	41 %	5 %	6 %
Other interactions and remainder	40 %	18 %	12 %

On the control the two appear as an interaction (41 %) because only one product of repeats and scale is right, a total of 2. That is the same statement in another form.

The kind of loading matters little once the scale is right. At scale 0.5 the three loadings are level: 65.1, 66.1 and 62.4 points on Taiwan, 59.5, 59.8 and 58.7 on German Credit. The large loading range on Taiwan in Figure 1 is the IQP loading at the default scale (36.5 points, with 22.1 points between its two starts): this loading needs a gentler scale than the other two, and at 0.5 it is back in line.

What this means for you. Treat loading repeats and scale as one budget. Raise one or the other, never both, and there is no need to choose a richer loading.

4. Finding 3: the method sees a signal where there is one

On the teacher control the default scale is too gentle by a factor of two, and the classifier sits at chance (35.0 points). At the right scale it reaches **78.6 points**, with both random starts within 0.3 points of each other. The product's standard models on the same rows stay between 33.3 and 41.7 points. The best classical reference we have for this file, a model that is given periodic features, stands at 64.0: our earlier measurement on the same file (study 8 of the series, registered; publication to follow).

This matters for everything else in the study. It shows that the instrument is not blind: when a dataset holds a pattern of the kind a quantum circuit expresses naturally, the classifier finds it, clearly and reproducibly. The modest scores on the two credit datasets are therefore a statement about those two datasets. Whether a reader's own data looks more like the credit tables or more like the control is an open question, and section 9 shows how to settle it in an afternoon.

One detail is worth a sentence. The control's labels were made with the richer ZZ loading, and still the plain loading at the right scale (78.6) beats the ZZ loading at its best (58.3). The richer loading is not needed even on data that was made with it.

What this means for you. A weak result on your data is information about your data, not a failure of the tool. The data is the largest factor, the scale the second, and the test that tells you is cheap.

5. Finding 4: the model can learn the dial

The map says that the scale decides and that the right scale depends on the data: gentle for the credit datasets, wide for the control. A search finds it, at the price of many trainings. So we built the obvious answer into the product and measured it: **trainable input scaling**.

With the option on, the training learns one scale factor and one offset for every feature, starting from the configured scale. The circuit stays the same size; the model gains two trained numbers per qubit, and each factor is limited to between one third and three times the start.

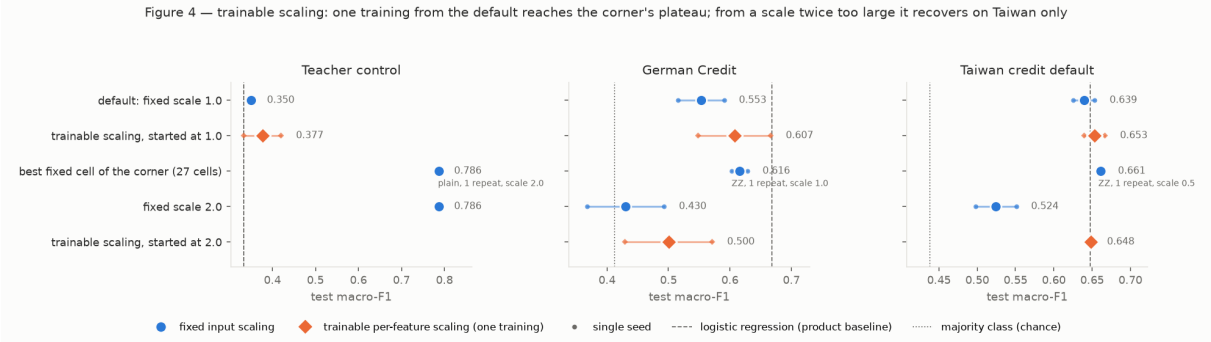


Figure 4. The default, trainable scaling started from the default, and the best corner cell.

	Default	Trainable scaling, from the default	Best of the 27 corner cells
German Credit	0.553	0.607	0.616
Taiwan credit default	0.639	0.653	0.661
Trainings needed (two random starts)	2	2	54

One training from the default lands within one point of the best configuration that the search found, on German Credit, on Taiwan and on the internal dataset.

How to read this result. The gain over the default (5.5 points on German Credit, 1.4 on Taiwan) is inside the noise level of each dataset, so no single table proves it. The evidence is that the option lands at the same place three times: next to the plateau that the search had located independently. The claim is about saving a search, not about a higher ceiling. Trainable scaling never beat the best fixed scale.

What the model learns. The learned factors say why one training can replace a search. On Taiwan the median factor is 0.62 and 0.75 in the two starts: the model moves towards the gentler scale where the plateau is. The factors also differ strongly between features, from 0.39 to 2.18. A single scale for all features cannot express that; the option can. The product's report lists the learned factor of every feature.

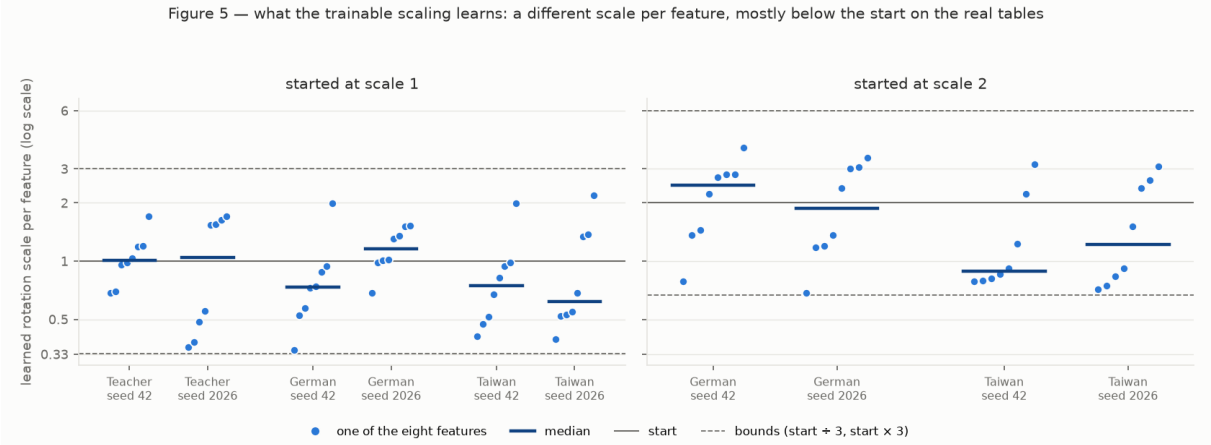


Figure 5. The learned scale factor of each feature, per dataset and random start.

What the option does not do. It refines a starting point; it does not search.

- From a start that is too wide. Started at scale 2.0, it recovers fully on Taiwan (64.8 points against 52.4 for the fixed scale) and only part of the way on German Credit (50.0 against 43.0). The factors follow the slope of the training loss, and on a small dataset training stops before they have travelled far enough.
- From chance. On the control the default scale gives a score at chance, and the option ends at 37.7 points with factors that have barely moved. Where there is no slope there is nothing to follow.

What this means for you. Start from the default scale and switch the option on: one run does the work of a search over 54. If the default itself is at chance, try scale 0.5, 1 and 2 once before concluding anything.

6. Finding 5: the cheapest circuit is the right one

Two-qubit gates are the operations that tie qubits together. On a quantum computer they are the slowest and the most error-prone, so their number is the price of a circuit. The default circuit has 14 of them. On no dataset does a circuit with more gates beat it by more than the noise level.

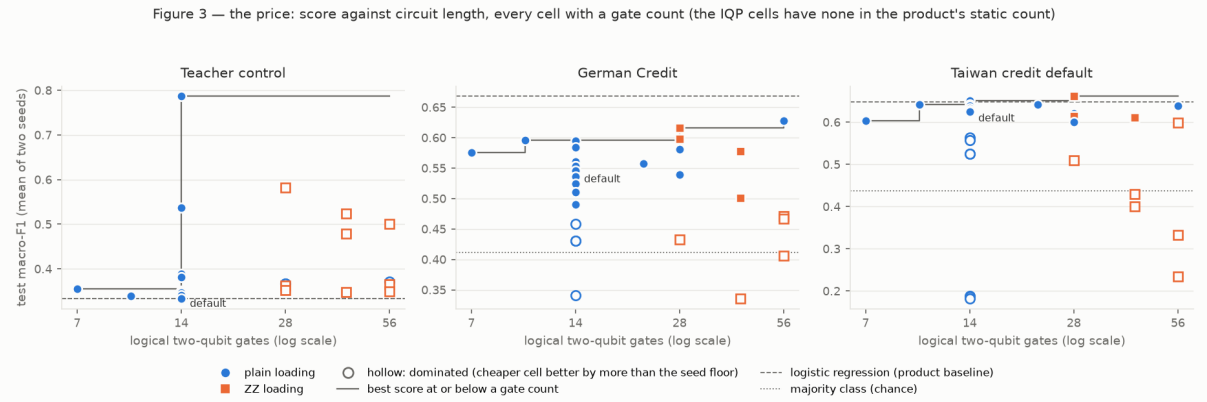


Figure 3. Score against the number of two-qubit gates of the trained circuit.

Change against the default	Two-qubit gates	German Credit	Taiwan credit default
Default (plain loading, linear entanglement, 2 layers, 8 qubits)	14	0.553	0.639
ZZ loading	28	0.616	0.613
Full entanglement	56	0.628	0.639
4 layers	28	0.539	0.620
12 qubits	22	0.557	0.642
6 qubits	10	0.596	0.641

On German Credit the ZZ loading and the full entanglement show gains of 6.3 and 7.5 points. Both are inside that dataset's noise level of 12.1 points, and Taiwan, the more precise instrument, shows no gain for either. With a third random start on German Credit the plain loading at scale 0.5 leads there too (59.7 points).

Six qubits score what twelve score: 64.1, 63.9 and 64.2 points on Taiwan for 6, 8 and 12 qubits. Among the 27 corner cells, 5 of the 9 ZZ cells on German Credit and 6 of 9 on Taiwan are beaten by a cheaper circuit by more than the noise level. The default circuit is beaten by no cheaper one on any dataset.

What this means for you. Start small. The cheapest circuit is as good as any on these data, and fewer gates mean less exposure to errors on a device. Whether the scores hold on hardware is the question of the next study.

7. Finding 6: what you can leave alone

Each of these was measured with two random starts against the default. None moves the score beyond the noise level on any dataset. Read the table as a list of things that need no tuning.

Setting or lever	Largest change, German Credit	Largest change, Taiwan	Why it does not help
Re-uploading	−0.017	−0.014	Loading the data again between layers adds scale, not new information. Where the scale is right it has nothing to add.
Entanglement, linear or full	+0.075	−0.001	Linear entanglement already connects every qubit to the others through the chain. Full entanglement quadruples the gates for the same reach.
Processing layers, 1, 2 or 4	+0.023	−0.036	Two layers can turn every qubit freely. A third and fourth have nothing left to do; a single layer is slightly short on Taiwan.
Qubits, 6, 8 or 12	+0.043	+0.003	More qubits carry more of the data's secondary directions. Between 6 and 12 of them, at the default scale, that adds nothing. A much wider input is a separate question (section 11).
Richer readout (measuring pairs of qubits as well)	−0.007	−0.001	The standard readout already extracts what the circuit offers. The limit sits before the readout, in the scale.
More training rows (6,000 and 18,000 instead of 2,000)	not applicable	−0.012 and +0.000	The score is set by what the circuit can express at the given scale, not by the number of rows. Nine times the data buys nothing.
Ensemble of two restarts	+0.007	−0.001	Two restarts land in nearly the same place. Averaging them changes little.

Circuit families. Four of the product's five families are level: on Taiwan they score between 62.9 and 64.2 points. One family is different. The basic entangler scores 56.3 points on Taiwan, 7.6 below the default and outside the noise level. After the data is loaded, the trained part of the circuit has to turn each qubit so that the measured direction separates the classes. Four families can turn every qubit freely enough, and which gates they use for it is a matter of spelling. The basic entangler can only turn about one fixed axis and must measure straight after the entangling gates, so some directions stay out of reach. The number of trained values is not the explanation: the real-amplitudes family trains fewer of them than the default and is level.

© qubit-lab.ch

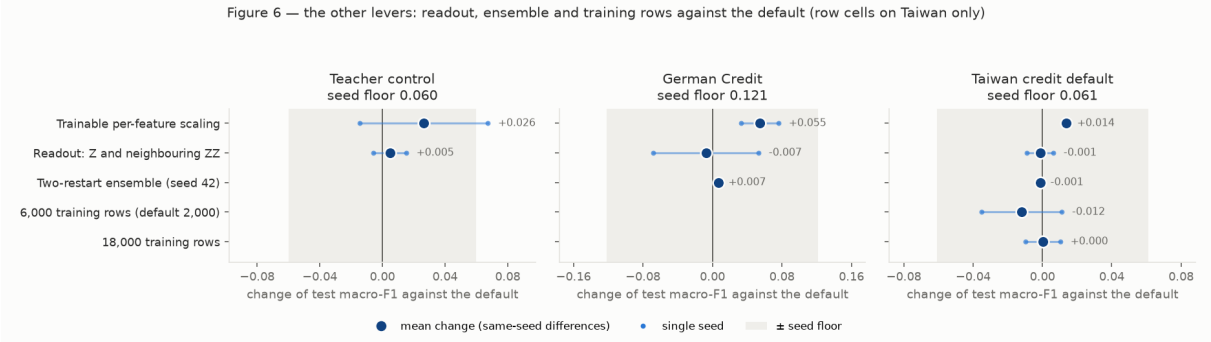


Figure 6. The levers of stages 2 and 3 against the default and the noise level.

What this means for you. Leave these at their defaults and avoid the basic entangler. Time is better spent on the data and its features than on the circuit.

8. Finding 7: the random start is a setting

	Teacher control	German Credit	Taiwan credit default
Configurations	35	36	36
Median difference between two starts	0.028	0.059	0.021
Noise level (four out of five configurations stay under it)	0.060	0.121	0.061
Largest difference between two starts	0.143	0.201	0.221
Settings whose range is under the noise level	6 of 8	6 of 8	5 of 8

Running the same configuration again changes the score by more than most settings do. This is useful to know, and it cuts in the reader's favour: a single training that "beats" another by two or three points, in our work or in anyone's, is not evidence yet. A second random start costs one more training and turns an anecdote into a measurement.

Three practical points follow from the data:

- **The variation is largest at the edge.** The widest differences sit where the scale is nearly too wide and one start trains while the other stalls. On the plateau the variation is small.
- **Small test sets widen it.** German Credit, with 200 test rows, has twice the noise level of Taiwan, and a third random start widened the range of results again. On a table of 1,000 rows the order inside the plateau cannot be measured; the plateau itself is stable.
- **A second start is worth more as a measurement than as an ensemble member.** Averaging the predictions of the two starts gains a median of 0.8 points on Taiwan over the mean of the two and nothing over the better one.

What this means for you. Always run twice. Read any comparison of two configurations, including a vendor's, against the difference between two runs of the same one.

9. Try it on your table

The map reduces the search to a short recipe in the QML RQP Pro. Each step is one run of the product.

Step	Runs	What you learn
1. Default configuration, two random starts	2	Your score at the default and your own noise level. Read every later difference against it.
2. The same with trainable input scaling on	2	Whether your data wants another scale. The report lists the learned factor per feature.
3. Only if step 1 is at chance: scale 0.5, 1 and 2, one start each	3	Whether a scale exists at which the classifier trains, as on the control.

How to read the result:

- **Step 2 lands above step 1 by more than your noise level:** your data wanted another scale, and the model found it. The learned factors show which features carry it.
- **Median factor clearly below 1:** your data wants a gentler loading, as the credit datasets do. A factor at its limit means the model wanted to go further; restart from 0.5 or 2.
- **Steps 1 and 2 agree within the noise level:** the default scale already suits your data. Leave the other settings where they are; the map says they will not change the answer.
- **Everything at chance, at every scale:** the features that reach the circuit do not carry the label in a form the classifier can use. More layers, more qubits or a richer loading will not change that; different features might.

Keep the plain loading, linear entanglement, two layers and eight qubits. Do not raise the scale and the loading repeats together. Compare the result with the standard models that the product reports in the same run.

At eight qubits one training took between five minutes and an hour in this study, depending on the size of the dataset.

10. Where the classifier stands against standard models

Stated once, for context. The product reports standard models on the same prepared rows in every run.

	Best configuration of this study	Logistic regression
German Credit	0.616	0.668
Taiwan credit default	0.661	0.648
Teacher control	0.786	0.333

On Taiwan the classifier is on par with logistic regression: the difference of 1.3 points is inside the noise level. On German Credit it is below, and on the internal dataset it is below as well. On the control it is far ahead of the product's standard models, by construction of the data. The study does not change any verdict of the product and makes no claim of an advantage on real data.

11. Limits

- **Simulation without noise.** Every score is a simulator value. Which configuration keeps its score on a device is the question of the next study.
- **Four datasets.** Two public credit datasets, one internal fraud dataset and one synthetic control. The map describes these; a reader's dataset is measured by the recipe of section 9.
- **Two random starts.** Enough to measure the variation, not enough to rank configurations inside a plateau. The gain of trainable scaling over the default is inside the noise level on each dataset.
- **One training recipe.** A different learning rate or patience may move the edge of the plateau and the reach of trainable scaling.

- **Eight features reach the circuit.** On the real datasets these are combinations that carry 52 % to 76 % of the selected features' variance. The standard models see the same eight. A pilot outside the product (design file, A23) suggests that this is the next limit after the scale. With every prepared column on its own qubit, and the qubits wired only in small groups or not at all, the classifier came within the noise level of logistic regression on the same columns on all three real datasets, again only at the right scale. A standard model built from the same sine curves matched it, so this is a finding about the input and not a quantum effect. On the control the picture is the reverse: there the wired circuit is needed, and both unwired qubits and the sine model stay far below it. The pilot's numbers are not product runs and are not part of this study. How wide the input should be, and when the wiring starts to matter, is the question of a follow-up study.
- **The price is the logical gate count.** Fitting a circuit to a device's layout adds gates and is not counted here. The product gives no gate count for the IQP loading, so its cells are not placed on the price axis.
- **Circuit families.** The basic entangler differs from the level families in three ways at once (few trained values, one rotation axis, no final rotation). The study does not separate them.

Scope statement (written before the first run). The results apply to the QML RQP's variational classifier as deployed on 6 November 2026, on the four datasets, the eight settings and the levels tested, as simulated without noise. They are a map of what moves the simulator score on these data, not a ranking of quantum classifiers in general, not a claim about a client's dataset, and not evidence about hardware. Every score is a simulator value; no verdict of the product and no Quantum Advantage Factor is changed by this study.

Appendix A. The datasets in detail

	German Credit	Taiwan credit default	Teacher control	Internal dataset
Kind	real, public	real, public	synthetic, made by a quantum circuit	real, fraud detection
Rows / features	1,000 / 20	30,000 / 23	1,000 / 8	large; sub-sampled
Positive class	bad loan, 30.0 %	default next month, 22.1 %	label 1, 50 %	fraud, about 1 %
Split (train / validation / test)	600 / 200 / 200	18,000 / 6,000 / 6,000	600 / 200 / 200	not shown
Rows the classifier trains on	600	2,000 (the product's cap)	600	not shown
What reaches the 8 qubits	8 principal components of 25 selected features (52 % of their variance)	8 principal components of 25 selected features (76 %)	the 8 features as they are	8 principal components
Chance (majority class)	0.412	0.438	0.333	not shown
Logistic regression	0.668	0.648	0.333	not shown
Licence	CC BY 4.0	CC BY 4.0	in-house	non-commercial

German Credit

The Statlog German Credit data from the UCI repository: 1,000 loan applications to a German bank, each labelled as a good or a bad credit risk, with 30 % bad. The 20 features describe the applicant and the loan: the state of the checking account, the duration and the amount of the loan, the credit history, the purpose, savings, employment, age, housing and others. Seven are numbers and thirteen are categories.

It is in the study because it is the public benchmark that much of the quantum machine learning literature uses, so the results can be read against other work. It is also small, and that matters for reading it: the test split has 200 rows, of which 60 are bad loans. One row more or less moves the score by about half a point, and two trainings of the same configuration differ by 5.9 points in the median. German Credit shows what a map looks like when the data is scarce.

The product prepares it with its defaults: with the categories coded as indicator columns the table has 61 columns, the 25 most informative are selected, and eight principal components of those go to the eight qubits. These eight numbers carry 52 % of the variance of the 25 features, so about half of the selected information never reaches the circuit. The standard models of the product see the same eight numbers.

Taiwan credit default

The "default of credit card clients" data from the UCI repository: 30,000 credit-card customers of a Taiwanese bank in 2005, labelled by whether they defaulted on the next monthly payment (22.1 % did). The 23 features are the credit limit, sex, education, marital status and age, and for each of six months the repayment status, the bill amount and the amount paid.

It is the study's most precise instrument. The test split has 6,000 rows, so the score of one training is stable to about one point, and effects of 3 to 6 points that German Credit cannot resolve become visible. Wherever the two public datasets differ in what they can show, Taiwan is the one to trust.

The preparation is the product's default, as for German Credit; the eight principal components carry 76 % of the variance of the 25 selected features. Of the 18,000 training rows the classifier trains on 2,000: this is the product's standard budget for a variational classifier. Four trainings of stage 2 lift that budget to 6,000 and to 18,000 rows to test whether more data helps (it does not, section 7). The standard models train on all 18,000 rows.

Teacher control

A synthetic dataset with a known answer: 1,000 rows of eight angle features, labelled by a quantum circuit (a "teacher") that loaded the features with the ZZ loading and two repeats. Half of the rows carry label 1. The file comes from the generator of study 6 (published); it is the same file our device-check study uses (study 8, registered; publication to follow).

The control is the instrument check. On real data nobody knows whether there is a pattern that a quantum classifier could find and a simple classical model could not. Here there is one by construction, and its scale is known: the labels depend on the features through rotations that are twice as wide as the product's default. So the control answers two questions that real data cannot: does the method see such a pattern at all, and does a tool that is supposed to find the right scale actually find it?

Unlike the real datasets, the control goes to the circuit raw: no feature selection, no standardisation, no principal components. Its features are already angles, and rotating them into principal components would destroy the pattern the teacher wrote into them. The first batch of 70 control trainings ran through the default preparation, ended at chance for exactly this reason, and was discarded and repeated (amendment A4). Logistic regression is at chance on the control (0.333), the product's tree-based models reach 0.355 and 0.417, and our earlier measurement on the same file (study 8, registered; publication to follow) recorded 0.640 for a classical model that is given periodic features.

The control is never compared with the real tables. Its numbers say what the instrument can do, not what a credit dataset should look like.

The internal dataset

A large real fraud-detection table with about 1 % fraud cases, used in-house since the advantage-island study. Its licence does not allow commercial use, so its numbers are not printed here. It ran through the same grid as the public datasets, with the product's option for heavily imbalanced data, which trains on a balanced sample and scores on the natural share. Where this study says "on every dataset" or "on every real dataset", the internal dataset is included; where a finding does not hold on it, the text says so.

Appendix B. Settings, recipe and definitions

The eight settings and their levels

Setting (name in the QML RQP Pro)	Levels tested	Product default
Loading method (**feature_map_type**)	plain (one rotation per feature), ZZ, IQP	plain
Loading repeats (**feature_map_repeats**)	1, 2, 3	1
Re-uploading (**data_reuploading**)	off, on	off
Input scaling (**input_scaling_bandwidth**)	0.5, 1.0, 2.0	1.0
Entanglement (**entanglement**)	linear, full	linear
Processing layers (**ansatz_reps**)	1, 2, 4	2
Qubits (**n_qubits**)	6, 8, 12	8
Circuit family (**ansatz_type**)	hardware-efficient, strongly entangling, RX-RY-CZ, real amplitudes, basic entangler	hardware-efficient

The names in brackets are the setting names of the QML RQP Pro, as they appear in its workbook. The text says "loading" for what the product and the literature call the feature map: the plain, ZZ and IQP loading methods are the product's **angle**, **zz_like** and **iqp** feature maps. Trainable input scaling is the setting **input_scaling_trainable**.

Training recipe, identical for every training. The recipe of the product's published sample workbook: Adam, learning rate 0.05, up to 600 iterations at batch 32, early stopping with patience 40, balanced class weights, the decision

threshold tuned on the validation rows. In a pilot with the product's shorter default recipe (50 iterations) the validation score was still rising when training ended, so the longer sample recipe was fixed before the first grid training (amendments A2 and A3). Exact expectation values, no shots, no device.

Definitions.

- **Score.** The product's macro-F1 on the test rows, with the decision threshold frozen on the validation rows. Macro-F1 averages the F1 of both classes, so a model that predicts only the majority class scores well below 0.5: 0.412 on German Credit, 0.438 on Taiwan, 0.333 on the control. These are the "chance" values. The second measure, PR-AUC, is in **analysis/tables.md**; on Taiwan it rises from 0.434 at the default to 0.469 with trainable scaling, against 0.473 for the best corner cell.
- **Cell score.** The mean of the two random starts (training seeds 42 and 2026) of a configuration.
- **Seed spread and seed floor.** The seed spread of a configuration is the difference between its two starts. The seed floor of a dataset is the 80th percentile of those spreads over all its stage-1 configurations: 0.121 on German Credit, 0.061 on Taiwan, 0.060 on the control. The text calls it the noise level. It was registered before the first run.
- **What the seed changes.** In the product, the training seed sets the starting values and also re-draws the feature selection. The noise level therefore measures the whole of "the same configuration, run again", which is what a practitioner experiences.
- **Range of a setting.** The highest minus the lowest cell score across the levels of one setting, with the other seven at their defaults.
- **Beaten on price.** A cell is beaten when another cell of the same dataset has fewer two-qubit gates and a score higher by more than the noise level.

Appendix C. The registered hypotheses

Six hypotheses were written down before the first run. Each verdict follows the registered rule: a comparison against the seed floor, with no correction.

	Hypothesis	Verdict	In one line
H1	Scaling and qubits together move the score more than loading, repeats and re-uploading together	not supported	Holds on the internal dataset only. The hypothesis put the loading repeats on the wrong side.
H2	Richer loadings win on the control and not on real data	not supported	The real-data half holds on every dataset. The control half fails: the plain loading wins there too.
H3	Three loading repeats are never better than two on real data	not supported	Holds on Taiwan. Fails on German Credit on one pair of cells that did not train properly.
H4	Re-uploading helps with one or two layers and not with four	not supported	Re-uploading gained more than the floor on no dataset at any depth.
H5	The seed spread exceeds the effect of at least four of the eight settings	supported	6 of 8 settings on German Credit, 5 of 8 on Taiwan; holds on the internal dataset too.
H6	The plain loading is on the price front and at least one ZZ cell is beaten on price	supported	The default cell is beaten by no cheaper one on any dataset.

Four of six are not supported by the letter. Two of them deserve a sentence, because the way they fail is the finding:

- **H1** treated the loading repeats as a structural setting and set them against the scaling. The data shows that repeats are scale (section 3). With repeats counted on the side of scale, the two together carry 72 % and 77 % of the corner. The hypothesis was wrong about the grouping, and the study is better for it.
- **H3** fails on German Credit because of one pair: the ZZ loading at scale 2.0 scores 0.336 with two repeats and 0.471 with three. Neither cell trained properly (chance is 0.412), and the difference of 0.135 is barely above the floor of 0.121. No cell that trains gains from a third repeat.

H2 and H4 fail in the direction of "even less than expected": the richer loading does not win even on the control, and re-uploading helps nowhere.

Appendix D. The decision table and the product

The storyline of 6 November held four result patterns with a recommendation each, written before the experiment. The row that applied:

Result pattern	Client recommendation	Product wording
A few settings explain most of the spread, the same ones on both real datasets	Tune those, leave the rest at default	Planning step orders the settings by measured effect; "within randomness" note for the rest

The two settings are the input scaling and the loading repeats, and they are one dial. The third pattern of that table ("the seed spread swallows most settings") describes the same data from the other side and is true as well: 5 to 6 of 8 settings are under the floor.

What the product changes with this study.

1. **Trainable input scaling** is available as an option of the variational classifier. The report shows the learned factor per feature, and the code exports and the device check carry the learned scale with the model.
2. **Settings in the order of their measured effect.** The planning step says which settings moved the score in this study and which did not, and on the page the scale controls come first in the parameter card.
3. **A note on training randomness.** The report's performance section and the user guide say how far two trainings of one configuration differed in this study, and recommend a second random start before two configurations are compared.
4. **Warnings and hints in the planning step.** A warning when loading repeats times scale reaches 2 or more (warned, not refused: the control needs it), and hints for the IQP loading, the basic entangler and a short training budget.

No default was changed. The gains of a gentler default scale and of trainable scaling are inside the noise level on each dataset, and the rule written before the study was "no number, no change". The four changes are live since 9 November 2026 (product image **backend-20261109-study9-replay**).

Appendix E. Record

- **Series.** Study 9 of the qubit-lab.ch evidence series (QL-ST-009), November 2026.
- **Design.** Locked and committed on 7 November 2026 before the first grid training. Every later change is a dated amendment in **DESIGN.md** (A1 to A25), including the change of the training recipe (A3), the correction of the control's data path (A4), the levers of stage 2 (A13) and the three further circuit families (A16). The amendments also record the pilots that are not part of the study (A11, A23, A25).
- **Runs.** 360 trainings in **runs/ledger.csv**, each with its configuration, seed, run id, code version and scores. Stage 1 ran on the pinned product image and, for part of the grid, locally through the same code. Stages 2 and 3 ran on the product build that adds the two new options; that build reproduces the pinned code bit for bit on the cells that use no new option (A12b), and four cells were trained again on the final code with identical predictions (A19).
- **Definitions.** The seed floor is the registered 80th percentile of the seed spreads. Some working notes in the amendments quote the largest spread instead; the exhibits and this study use the registered definition throughout (A18).

- **Exhibits.** **analysis/tables.md** (Tables 1 to 6), **analysis/hypotheses.md**, Figures 1 to 6, all generated by **analyze.py** from the ledger.
- **Reproducing a cell.** Every cell is a product run with settings that the product accepts in its workbook or as overrides. The datasets are public (German Credit, Taiwan) or generated by a published recipe (the control).

Disclaimer

This document reports a method study on two public credit datasets, one internal dataset and a synthetic control with a rapid quantum prototyping tool by qubit-lab.ch. Its figures are unedited outputs of the described runs, intended for technical review, education and structured discussion. The settings found to move the score or not to move it are properties of the tested configurations on the tested datasets under one training recipe; they are not financial advice, not a performance claim for other datasets, and not a hardware result.

Legal notice. © 2026 qubit-lab.ch. All rights reserved. This document and the software it describes are proprietary to qubit-lab.ch and provided for authorized use only; reproduction or redistribution without written permission is prohibited. Outputs of the RQP tools are intended for technical review, experimentation, demonstration, and structured discussion - they are not financial advice. Full legal terms: qubit-lab.ch/legal