Four-Qubit Quantum Kernel Methods for Wildfire Modeling

Quantum Bears 1 | Project report

Josh Borthick · Kyle Keltner · Isaac Smith
Asher Van Maanen · Jonathan Block
Missouri State University · Springfield, Missouri, USA
Department of Physics, Astronomy and Materials Science (PAMS)
Department of Computer Science (CS)
Career level: Undergraduate
Submission category: Rising-star
Track: Quantum Machine Learning
October 24, 2026

Abstract

We adapted a three-qubit classroom demonstration to wildfire modeling on Los Angeles County land-cover data. A four-qubit kernel supplies distances to a five-neighbor regressor for hourly danger classification and a separate fuel-estimation/spread experiment. Across 48 snapshots, quantum KNN achieved mean held-out macro-F1 of 0.6866, versus 0.6713 for the original forecast and 0.7260 for same-feature Euclidean KNN. Its hour-48 reached area was closer to the reference, but its future-spread F1 was lower. A complementary classical RBF-SVM classified static susceptibility with 85.3% random-holdout accuracy and 37.4% spatial-block accuracy. Together, the experiments demonstrate a small-qubit hybrid workflow and the importance of spatial evaluation, without establishing a quantum advantage.

Summary of contributions

- Extended the lecture feature map from three to four qubits and replaced SVM classification with kernel-distance KNN regression.
- Integrated real geographic land-cover data with a 90 m grid and barrier-aware simulated fire travel times.
- Created 48 hourly datasets with 200 points each, preserving spatially separated training and test partitions.
- Compared quantum and Euclidean KNN; added a classical RBF-SVM susceptibility experiment.
- Added a separate fuel-estimation model, three comparable spread simulations, diagnostic plots, and 49-frame animations.

1. Motivation and project development

Our motivation is to anticipate how threatened areas change as a fire progresses. Similar to weather forecasting, a time-dependent map could help communicate when locations may face increasing danger and, after appropriate validation, support evacuation planning. This project explores that goal through a controlled simulation benchmark rather than an operational warning system.

We chose quantum machine learning to investigate whether a compact quantum representation could help a simple classical estimator distinguish spatial and temporal patterns. Kernel methods compare observations through their representations; our circuit changes the similarity measure used to select KNN neighbors. Classical nonlinear kernels can also provide expressive representations, so improvement must be established by comparison rather than inferred from the use of qubits.

Development from the supplied lecture example

Aspect	**Lecture notebook [1]**	**Team implementation [2]**
Data	200 synthetic points; two features; four classes	LA County grid; 48 snapshots; four input features; five danger classes
Quantum circuit	Three qubits; RX/RZ, CZ ring, RY, interaction RZ	Four qubits; same gate sequence extended to four wires
Estimator	SVC with a precomputed fidelity kernel	K=5 inverse-distance KNN regression; rounded class outputs
Evaluation	Random 80/20 split; hinge loss	Preserved spatial holdout; matched Euclidean control; hourly metrics
Outputs	Decision zones, circuit, SVM coefficients	Kernel/distance matrices, tables, maps, confusion matrices, spread animations

The supplied lecture code assigns fire-zone names to synthetic classes. Our contribution is the geospatial and temporal application, the four-qubit adaptation, the KNN feedback workflow, and the physical propagation branch. The recorded lecture scores are not used as a wildfire baseline because the datasets and targets differ.

2. Geographic data and simulation assumptions

The numerical input is LA_County_Fire_Susceptibility_0_to_4.kml. Its embedded metadata identifies a custom reclassification of USGS Annual NLCD 2024, Collection 1.1, with Census county geometry and areal hydrography [3–5]. The KML describes aggregation from 30 m pixels to 90 m modal classes and removal of small nonwater patches. The notebook rasterizes the polygons in UTM Zone 11N (EPSG:32611), preserving water and unknown coverage.

This layer is a land-cover screening index, not an official CAL FIRE hazard-severity map or measured probability of burning. The separately supplied Google Earth overlay files are contextual resources; the final notebook's numerical input path points to the scored LA County KML.

Score	Land-cover grouping	Speed (m/s)	90 m crossing
0	Water	0	Blocked
1	Developed / barren	0.02	75 min
2	Grass / crops / open space	0.10	15 min
3	Shrub / scrub	0.25	6 min
4	Forest	0.50	3 min

These fixed speeds are project assumptions. The grouping also includes wetlands and other NLCD classes as documented in the KML; it does not imply that forests are universally more hazardous than chaparral. Wind, slope, moisture, suppression, and ember transport are not modeled.

Fire propagation uses Dijkstra shortest-path travel times on an eight-neighbor cell graph. For adjacent cells i and j, edge time in hours is $d_i \cdot (1/v_i + 1/v_j)/7200$, where distance is in meters and speed is in meters per second. Water and unknown cells are excluded; diagonals cannot cut blocked corners. All three simulations use the same saved ignition point.

Danger classes are project-defined arrival-time bands

At issue hour h, remaining arrival time determines the class: Extreme (red), ≤3 hours; Very High (orange), >3–6 hours; High (yellow), >6–12 hours; Moderate (blue), >12–24 hours; Low (green), >24 hours or no modeled arrival within that window. These are not calculated NFDRS ratings. Low danger is distinct from fuel score 0, which denotes water. Arrivals are calculated through hour 72 to retain a full 24-hour look-ahead at hour 48.

3. Four-qubit representation and KNN

Each hourly model uses four ordered inputs: projected easting, projected northing, the baseline continuous susceptibility estimate, and baseline remaining arrival time. Remaining time is clipped to [0,48] hours; an infinite arrival receives the value 48 as a numerical sentinel. Hour is metadata, not a fifth circuit input. Separate models and training-only MinMax scalers are fitted for each hour.

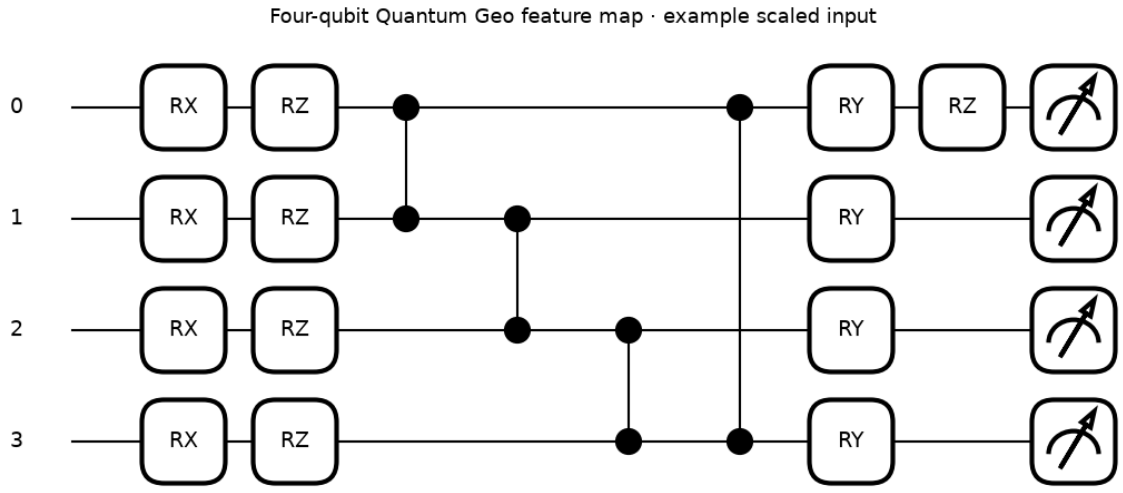


Figure 1. Four-qubit adaptation of the lecture feature map. The circuit image is extracted from the submitted notebook.

Starting from $|0000\rangle$, the feature map applies $RX(\pi x_i)$ and $RZ(\pi x_i)$ on each wire; a CZ ring connects 0–1–2–3–0; $RY(\pi x_i^2)$ follows on each wire; and $RZ(\pi x_0 x_1)$ acts on wire 0. The resulting state contains 16 complex amplitudes. This is angle encoding, not direct amplitude encoding of the four inputs.

The fidelity kernel is $K(x,z) = |\langle\varphi(x)|\varphi(z)\rangle|^2$. Kernel similarities become KNN distances through $D(x,z) = \sqrt{\max(0, 2 - 2K(x,z))}$. The regressor uses five neighbors with inverse-distance weighting, then rounds and clips predictions to the allowed danger classes [6,7].

The implementation calculates exact state vectors on a classical computer and checks its vectorized circuit against PennyLane. There are no hardware measurements or shot noise in the reported run. The circuit defines a quantum feature map, but these results do not demonstrate computational speedup.

4. Workflow and reproducibility

1. Load the saved baseline arrays, manifest, and spatial holdout mask. Load the scored KML and verify grid alignment, masks, speeds, and the calculation horizon.
2. At each hour 1–48, sample 40 eligible locations per baseline danger category: 32 from training blocks and eight from held-out blocks. Use seed 2026 and no replacement within each hourly stratum.
3. Form the four input features and use reference-simulation danger codes as targets. Exclude currently reference-reached, water, and unknown cells from the hourly cohort.
4. Fit the scaler on 160 training records, encode all 200 records, and calculate the 200 × 200 kernel. Train with the 160 × 160 training block and evaluate with the 40 × 160 test-to-training block.
5. Fit the same-feature Euclidean KNN control with identical labels, scaling, K, weights, and training/test records. Record hourly metrics and confusion matrices for all three approaches.
6. Fit a separate quantum fuel regressor using the hour-1 features and 160 training locations, with source fuel scores 1–4 as its targets. Freeze that model and estimate one static fuel field.
7. Run the common travel-time solver on reference, classical, and quantum-estimated fuel fields. Export hourly results, maps, data sheets, and county/detail animations for hours 0–48.

Sampling totals are 9,600 hourly records: 7,680 training records and 1,920 held-out records. These are repeated hourly samples, not necessarily 9,600 distinct locations or 48 independent fires. Equal allocation across baseline zones does not guarantee equal reference-class counts or area-weighted county performance.

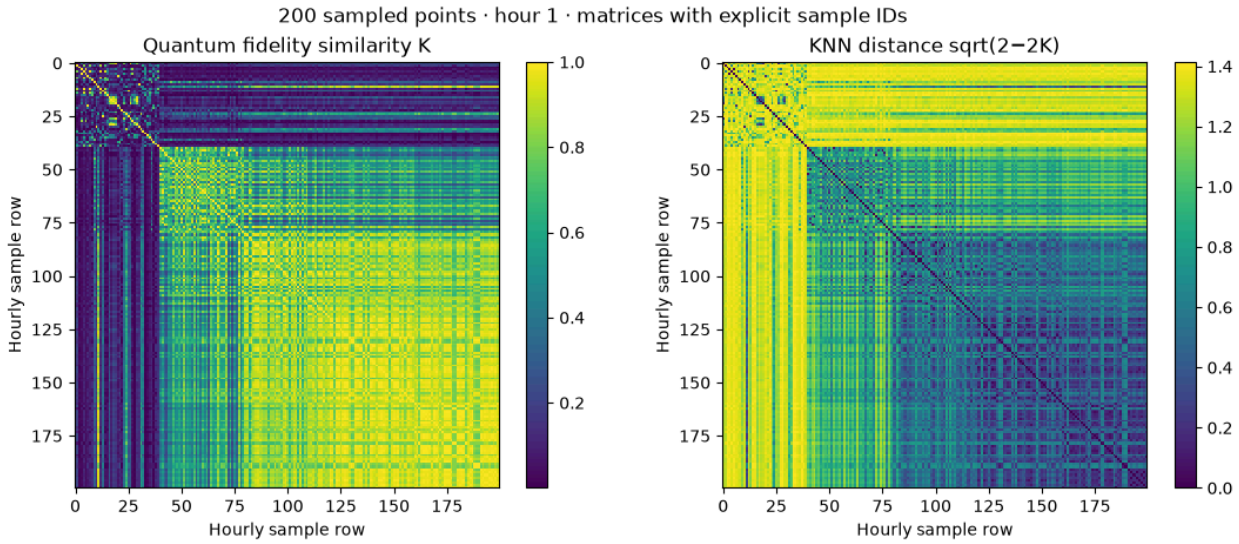


Figure 2. Hour-1 fidelity kernel and derived KNN distance matrix for the 200 sampled points. Training and test subsets are tracked explicitly.

5. Results: hourly danger classification

The primary hourly comparison uses 40 spatially held-out records per hour. Macro-F1 averages class-level F1 over all five codes; the table reports the arithmetic mean of those hourly scores. Values below come from the saved notebook outputs, not a new execution.

Method	Mean hourly macro-F1
Original KNN forecast	0.671276
Same-feature Euclidean KNN	0.726005
Four-qubit kernel KNN	0.686630

The quantum model improves on the original forecast by 0.015354 macro-F1, but trails the matched Euclidean control by 0.039375. The matched comparison is especially informative because only the distance representation changes. The results support feasibility with 160 training points per hourly model, but not overall superiority of this quantum feature map.

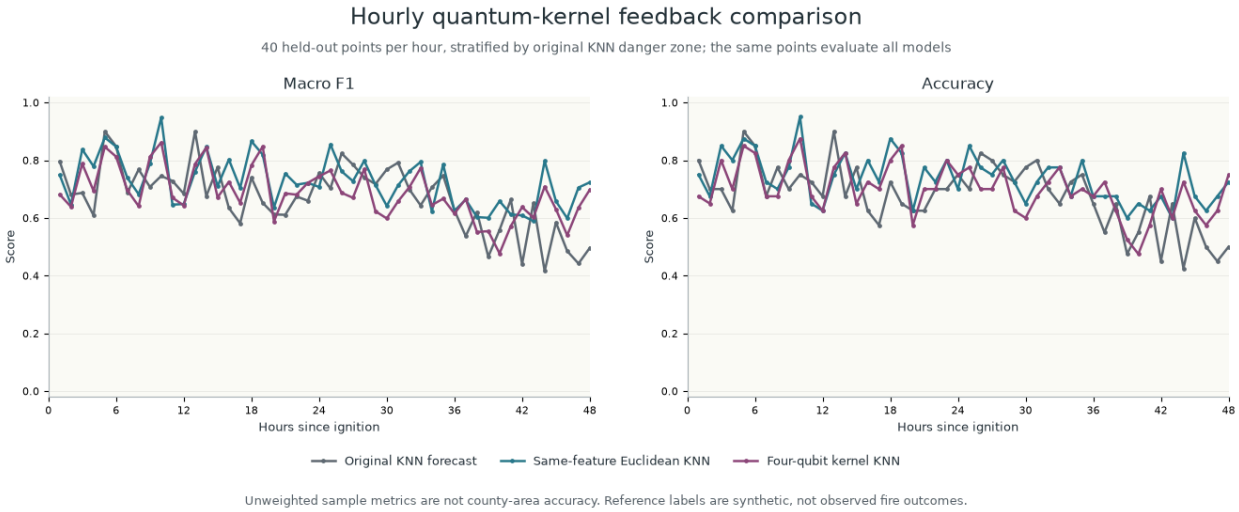


Figure 3. Held-out macro-F1 and accuracy across 48 hourly snapshots. All methods are evaluated on the same 40 sampled test points within each hour.

Performance varies over time. The team found it encouraging that a small four-qubit representation could capture useful structure in the simulated danger zones. A stronger temporal forecasting claim requires testing on unseen ignition scenarios or future times without fitting a new model using that hour's reference labels.

6. Results: class-specific behavior

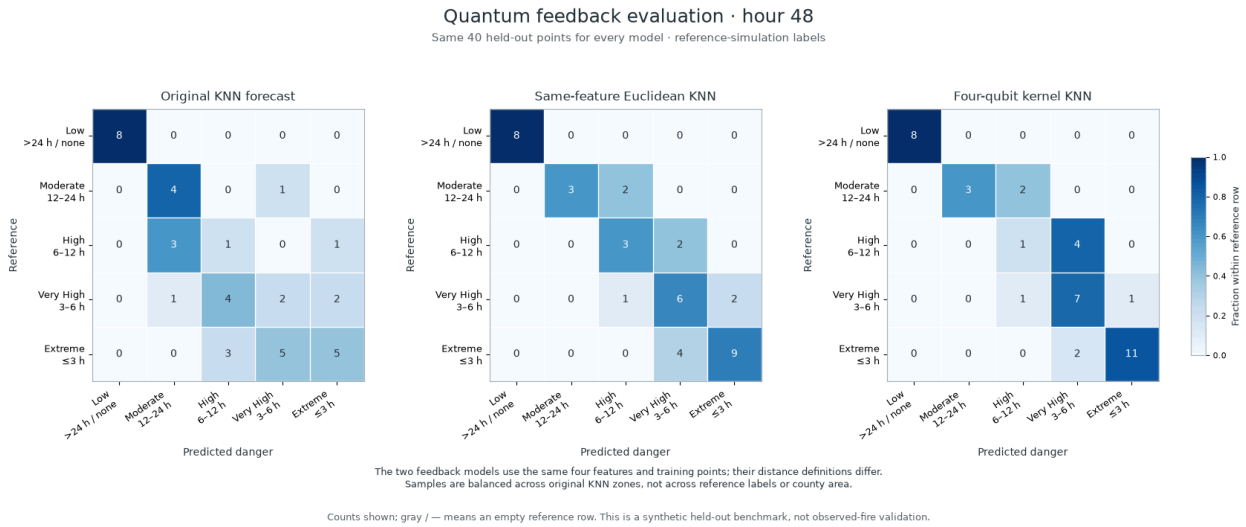


Figure 4. Hour-48 confusion matrices. Rows are reference classes and columns are predictions; numbers are counts and shading is normalized within each reference row.

At hour 48, the original forecast correctly classifies 20/40 points (50.0%), Euclidean KNN 29/40 (72.5%), and quantum KNN 30/40 (75.0%). The corresponding macro-F1 values printed for that hour are approximately 0.496, 0.723, and 0.698. Higher accuracy does not necessarily imply higher macro-F1: the metrics weight class performance differently.

For the Extreme reference class, the quantum model identifies 11 of 13 points as Extreme, versus nine for Euclidean KNN and five for the original forecast. However, quantum KNN correctly identifies only one of five High points, assigning the other four to Very High; Euclidean KNN identifies three of five correctly. This explains why the quantum model's higher total correct count coexists with lower macro-F1.

The reported matrices describe one sampled hour. They do not establish a consistent advantage for any particular class across all times. Sampling is balanced by baseline category, so the reference rows contain unequal counts.

Interpreting the two experiments

Hourly danger classification uses reference future-danger labels to train a separate model at each hour. The physical simulation instead freezes a fuel regressor from hour 1 and evolves the resulting travel-time field. These answer different questions and have different evaluation populations. Their F1 values should not be combined into a single accuracy claim.

7. Results: physically propagated fire

The fuel-regression branch predicts susceptibility once and preserves the known water/unknown masks. Each map shows its own cumulative reached footprint in gray, with colors indicating its own remaining arrival-time bands. “Reached area” does not mean the area is actively flaming at that instant.

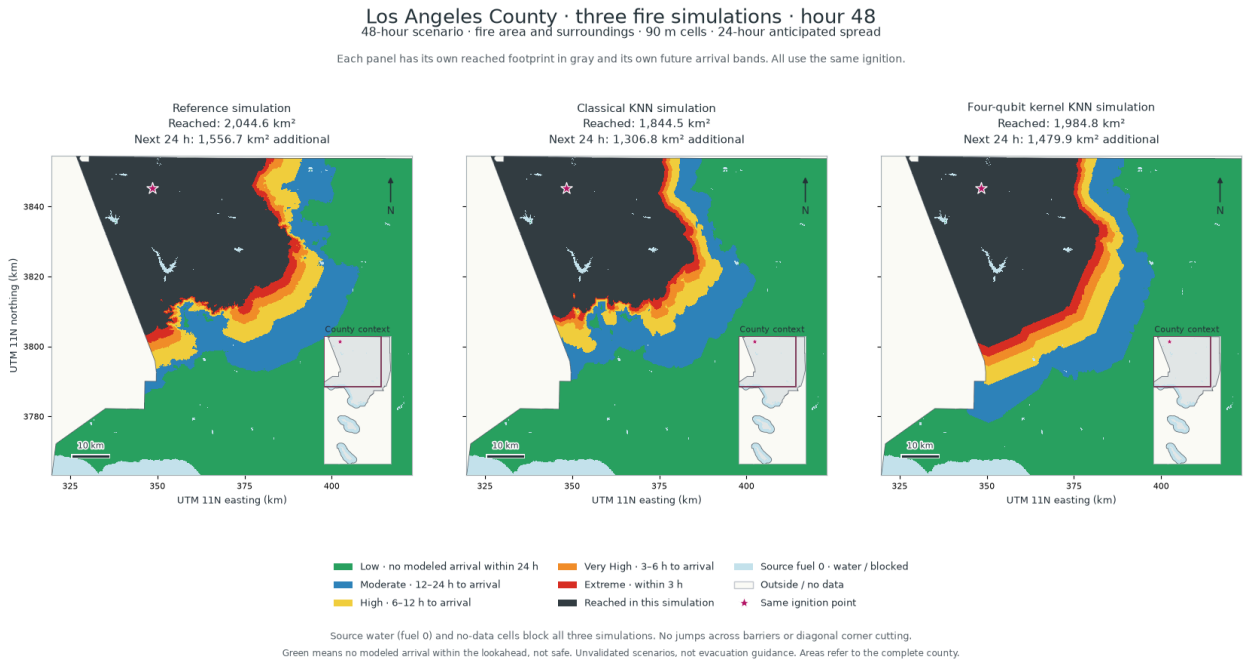


Figure 5. Reference, classical KNN, and quantum-kernel KNN simulations at hour 48. The expanded view shows differences in each model’s own reached footprint and future arrival bands.

Hour-48 quantity	**Reference**	**Classical**	**Quantum**
Reached area (km²)	2,044.59	1,844.52	1,984.75
Absolute area error (km²)	—	200.08	59.84
Relative area error	—	9.79%	2.93%
Next-24-hour binary F1	—	0.868022	0.796612

The quantum model is closer in total reached area at hour 48. However, the classical model has higher binary F1 for arrival within the next 24 hours on the fixed held-out, currently reference-unreached population. Area agreement measures size; it does not establish correct spatial overlap or better warnings.

8. Additional experiment: classical SVM

We also implemented a classical RBF-kernel support vector machine to classify static LA County susceptibility from location [8]. This complementary experiment learns the five source land-cover classes, including water. It does not use a quantum circuit or predict time-dependent danger, so its scores are not directly comparable with the hourly KNN benchmark.

Dataset and training

The notebook parses 6,027 polygons, respecting interior holes, and labels random interior points by polygon membership. With seed 42, 120,000 candidate locations yield 53,340 labeled points. Capping each class at 3,000 leaves 13,690 points: 3,000 each for scores 0–3 and 1,690 for forest. Longitude and latitude are converted to approximate local kilometer coordinates.

A StandardScaler–SVC pipeline uses an RBF kernel and balanced class weights. Three-fold stratified cross-validation searches $C \in \{1,10,100\}$ and $\gamma \in \{10,50,200,500\}$, selecting $C=1$ and $\gamma=500$ with balanced accuracy 0.845. The scaler is fitted inside the pipeline. No temporal, weather, terrain, or spectral features are included.

Measure	**Random holdout**	**15 km block holdout**
Training / test points	10,267 / 3,423	10,321 / 3,369
Accuracy	0.853	0.374
Balanced accuracy	0.857	0.391
Macro-F1	0.850	0.388
Cohen’s κ	0.815	0.213

The random split holds out 25% of points with class stratification. The spatial split holds out 20 of 83 occupied 15 km × 15 km blocks. Both models use the selected hyperparameters, but are fitted to their respective training subsets. Balanced accuracy is mean class recall.

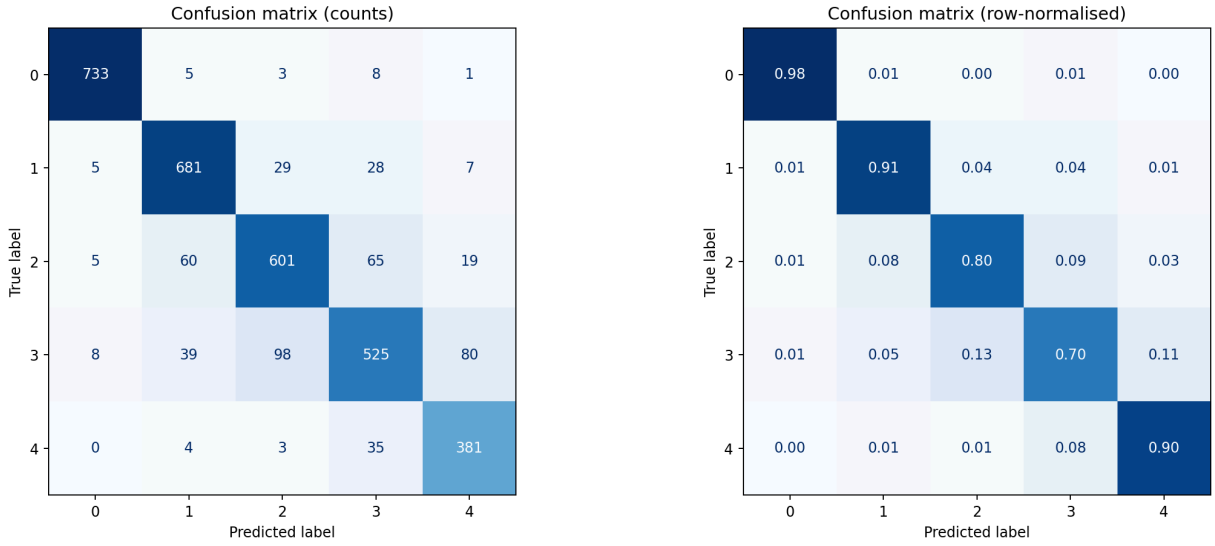


Figure 6. Random-holdout SVM confusion matrices: counts and row-normalized values. Codes 0–4 denote water, developed/barren, grass/crops, shrub/scrub, and forest.

Spatial predictions and interpretation

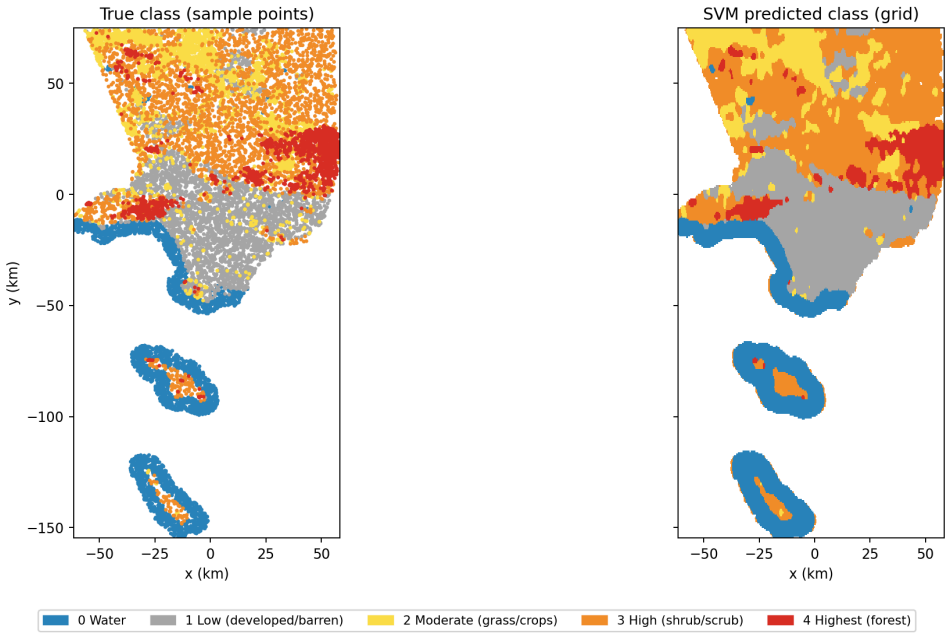


Figure 7. Sample labels and the final SVM susceptibility map. The map model is refitted on all 13,690 points; this visualization is not a held-out evaluation.

The final model predicts a 350 × 350 grid and hides locations more than 2 km from any sampled point. This proximity mask is an approximate coverage filter, not an exact county-boundary or water mask. The classifier can also accept new longitude/latitude pairs; those predictions require the same geographic-domain caution.

The random-split result shows that location alone can reconstruct much of the sampled class pattern. Water has recall 0.977 and forest 0.901 on that split, while shrub/scrub recall is 0.700. With whole blocks withheld, forest recall falls to 0.163 and developed/barren recall to 0.218. Shrub/scrub recall rises to 0.892 but precision falls to 0.228, indicating frequent assignment of other classes to shrub/scrub.

The large score drop is consistent with a coordinate-only model relying on nearby spatial examples rather than learning transferable environmental predictors. The two splits differ in geography and class composition, so the gap is not a controlled estimate of a single causal effect.

Evaluation limitation

Hyperparameters for the block model were selected on the separate random-split training pool, which can include points from the later held-out blocks. Thus the block-fit excludes those locations, but hyperparameter selection is not fully independent of them. A stricter follow-up would hold out outer spatial blocks before tuning and use spatial cross-validation within the remaining blocks.

SVM metrics and figures were generated by executing the supplied notebook with seed 42 and its stated settings. The execution environment used NumPy 2.3.5, pandas 2.2.3, Matplotlib 3.10.8, and scikit-learn 1.8.0. Existing quantum/KNN results retain the saved outputs described in Sections 5–7.

9. Dynamics, limitations, and next steps

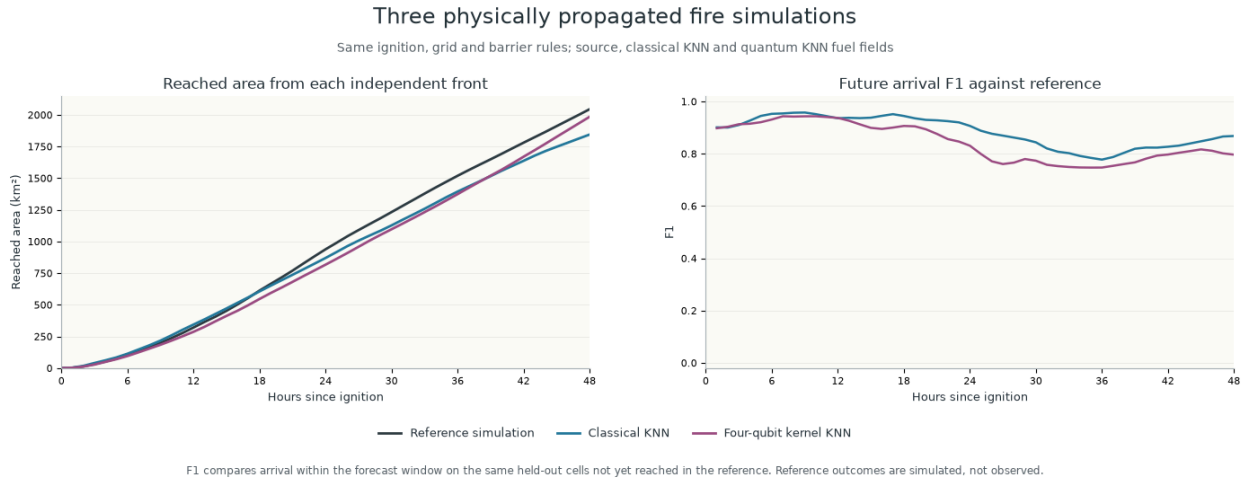


Figure 8. Reached-area trajectories and next-24-hour binary F1 for the physical simulations. Reference outcomes are simulated, not observed.

Saved checks confirm barrier preservation, monotonic reached footprints, and the speed bound. The two embedded 49-frame GIFs cover hours 0–48 and provide animated views of the simulations.

Limitations

One ignition scenario, synthetic outcomes, and 40 test points per hour limit generalization. Fuel-training labels are uneven: 107 of 160 records are shrub/scrub, while nine are forest. Results need replication across independent fires and sampling seeds.

Time enters through baseline ETA and separate hourly fits; the physical branch freezes an hour-1 fuel estimate. Simplified speeds and land cover constrain both approaches. More qubits alone would not resolve limited data or guarantee improvement.

Next steps

- Repeat independent ignition and sampling trials; report paired differences and uncertainty.
- Test temporal generalization and alternative classical/quantum kernels under matched tuning.
- Expand sampling and fuel-class balance; add environmental factors and observed-fire validation.
- Measure runtime, memory, shot-noise sensitivity, and hardware performance.

Conclusion

A four-qubit kernel can be integrated with a simple classical KNN model to represent time-dependent wildfire danger on real geographic data. The strongest result is a working, inspectable hybrid workflow with limited hourly training data. Mixed quantum performance and split-dependent SVM scores motivate further exploration and spatial validation, rather than a claim of quantum advantage.

10. Credits and references

Dr. Sakijda is credited for the original quantum-computing lecture notebook and presentation, including the feature-map structure and fidelity-kernel idea. OpenAI tools and agents were used to assist the project and preparation of this report.

[1] Dr. Sakijda. Quantum Geo lecture materials: EIDF_QUANTUM_GEO.ipynb and EIDF_QUANTUM_GEO.pdf. Course lecture materials.

[2] Quantum Bears 1. LA_County_Quantum_Geo_4Q_Physical_Simulation.ipynb. Project implementation and saved outputs supplied October 24, 2026.

[3] LA_County_Fire_Susceptibility_0_to_4.kml. Embedded metadata: custom land-cover reclassification, created October 24, 2026; USGS Annual NLCD 2024, Collection 1.1. Dataset documentation: https://data.usgs.gov/datacatalog/data/USGS%3A6810c1a4d4be022940554075

[4] Esri-hosted USGS Annual NLCD service. The supplied KML identifies the 2024 raster as ID 40.
https://di-nlcd.img.arcgis.com/arcgis/rest/services/USA_NLCD_Annual_LandCover/ImageServer

[5] U.S. Census Bureau, TIGERweb. County boundary and areal hydrography services cited in the supplied KML; county vintage listed as January 1, 2026.
https://tigerweb.geo.census.gov/arcgis/rest/services/TIGERweb/State_County/MapServer/1 and
https://tigerweb.geo.census.gov/arcgis/rest/services/TIGERweb/Hydro/MapServer/1

[6] PennyLane. Training and evaluating quantum kernels. Supporting documentation for quantum embedding kernels. https://www.pennylane.ai/demos/tutorial_kernels_module

[7] scikit-learn. KNeighborsRegressor documentation. Supporting documentation for distance-weighted regression with precomputed distances.
https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[8] Quantum Bears 1. LA_Fire_Susceptibility_SVM.ipynb. Classical RBF-SVM susceptibility classification, random holdout, spatial-block evaluation, and prediction maps. Executed with the supplied LA County KML and seed 42.

Software acknowledgments

The implementation imports NumPy and pandas for arrays/tables; SciPy for sparse graphs, Dijkstra travel times, and spatial operations; scikit-learn for scaling, neighbors, and metrics; PennyLane for circuit validation and drawing; Rasterio, Shapely, pyproj, and affine for geospatial processing; Matplotlib for figures; Pillow for GIFs; openpyxl for Excel export; ipywidgets and IPython for notebook playback; and threadpoolctl for thread control. Python standard-library modules support paths, XML parsing, hashing, metadata, and serialization.