QIOCE: VARIATIONAL QUANTUM INITIALIZATION FOR COUNTERFACTUAL EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating counterfactual explanations requires searching for prediction-changing inputs while balancing validity, proximity, sparsity, and plausibility. We propose QIOCE, a hybrid quantum–classical framework that learns direction-specific variational circuit parameters on training data and adapts them to each factual instance before classical refinement. Continuous and categorical features are represented through rotation and categorical-register encodings. We evaluate two complementary questions: the quality of valid counterfactuals and the ability of different initializers to reach a target class under shared refinement. QIOCE attains the lowest mean distance on five of seven tabular datasets in the valid-output comparison, and the highest or tied-highest validity on four of five datasets in the initialization study, with dataset-dependent validity–distance trade-offs. Handwritten-digit examples visualize nearby prediction-changing inputs and a second application as adversarial-like boundary probes. Measurement-based decoding and a small gate-based hardware experiment demonstrate an execution route beyond statevector simulation. Supplementary stage diagnostics reveal validity loss during quality optimization and demonstrate the effect of retaining valid candidates. The results support studying variational initialization while leaving matched-budget efficiency and robustness unresolved.

1 INTRODUCTION

A counterfactual explanation answers a concrete question: what changes to an input would lead a predictive model to a different decision? The generated input should reach the desired class while remaining close to the factual instance, changing few features and respecting the data distribution (Wachter et al., 2018; Mothilal et al., 2020). These goals can conflict: multi-objective formulations expose their trade-offs (Dandl et al., 2020), and benchmarking shows why proximity alone is insufficient (Guidotti, 2024). Mixed feature types add representation constraints, addressed by recent latent-variable and diffusion approaches (Panagiotou et al., 2024; Zhang et al., 2026).

QIOCE uses a variational quantum circuit to initialize counterfactual search. During training, it learns separate circuit parameters for the two class-transition directions. At inference, the corresponding parameters initialize an instance-specific optimization for the factual input, first targeting the desired class and then improving counterfactual quality. The decoded candidate is subsequently passed to a classical refiner that searches for a valid, nearby counterfactual. Figure 1 summarizes the full procedure.

Our evaluation separates counterfactual quality from target-class reachability. In the first comparison, all methods generate valid counterfactuals for all attempted instances, so proximity, sparsity, and plausibility are evaluated over the full population without conditioning on success. In the second comparison, failed attempts are retained and target validity is reported together with mean distance over all attempts. Reporting these quantities jointly prevents a low distance from being mistaken for a high-quality counterfactual when the target class was not reached.

The image examples that reconstructed digit can remain visually similar to the factual input while changing the classifier's prediction. Such an input constitutes a model-level counterfactual: it changes the model's decision without requiring the reconstructed image to be perceived by a human as belonging to the target digit class. With limited distributional guidance, the same framework

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

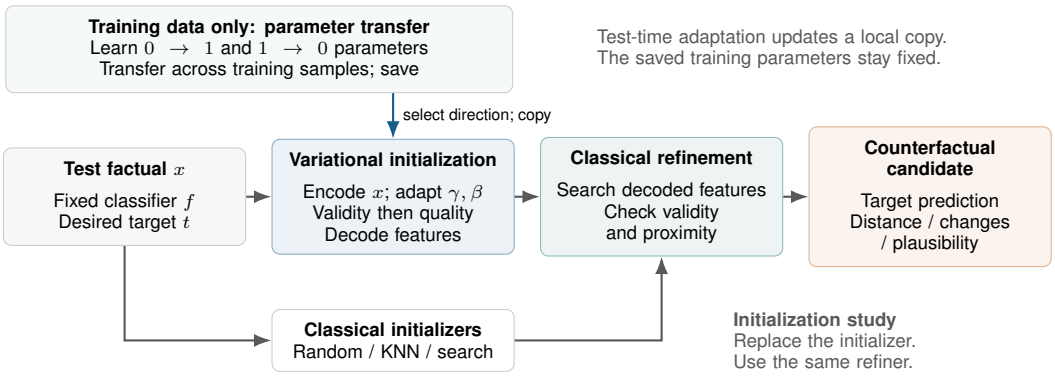


Figure 1: QIOCE separates training, instance-specific quantum search, and classical refinement. Training transfers parameters between training instances within each class-transition direction. Test-time adaptation uses a local copy of the saved parameters. In the initialization study, classical initializers replace the quantum search block and feed the same external refiner.

can also generate adversarial-like boundary probes. Both settings share the same prediction-change objective but differ in the emphasis placed on distributional plausibility.

Our contributions are: (i) a variational initializer with directional parameter transfer and instance-specific adaptation; (ii) tabular evaluations that separate valid-output quality from initializer performance under shared refinement; and (iii) image demonstrations and measurement-based decoding, including a small-scale quantum hardware experiment. We make empirical claims about counterfactual search performance, not computational quantum advantage.

2 Related Work

Search, feasibility, and recourse. Optimization-based counterfactual methods trade prediction change against proximity (Wachter et al., 2018). Growing Spheres searches for nearby prediction-changing alternatives (Laugel et al., 2017), DiCE promotes diversity among counterfactuals (Mothilal et al., 2020), and GeCo uses constrained genetic search (Schleich et al., 2021). Multi-objective formulations make competing criteria explicit (Dandl et al., 2020). Beyond proximity, actionable recourse constrains which features may change (Ustun et al., 2019), FACE favors density-supported paths (Poyiadzi et al., 2019), and prototype-based methods guide search toward class-representative regions (Van Looveren & Klaise, 2019). Causal recourse requires assumptions beyond changing a classifier's prediction (Karimi et al., 2021). QIOCE focuses on model-level counterfactuals and does not provide causal or hard actionability guarantees.

Generative counterfactuals and evaluation. Recent generative methods model the structure of the data distribution more explicitly. TABCF uses a transformer-based VAE for mixed tabular features (Panagiotou et al., 2024), DiCoFlex uses conditional normalizing flows to generate diverse counterfactuals with inference-time control (Furman et al., 2025), and tabular diffusion supports both continuous and categorical features (Zhang et al., 2026). CF-OPT extends counterfactual optimization to structured prediction (Vivier-Ardisson et al., 2024), whereas QIOCE considers a fixed binary classifier. Benchmarking frameworks such as CARLA, together with broader empirical studies, motivate evaluating counterfactuals along multiple dimensions rather than validity alone (Pawelczyk et al., 2021; Guidotti, 2024). Robustness to model changes constitutes a separate requirement (Jiang et al., 2024b); PROPLACE, for example, combines plausibility with certified robustness for neural networks (Jiang et al., 2024a). Our experiments evaluate counterfactual validity with respect to the fixed classifier and do not assess robustness after model retraining.

Variational quantum optimization. QAOA alternates cost and mixer unitaries (Farhi et al., 2014); the alternating-operator framework generalizes this design to constrained search (Hadfield et al., 2019). Variational algorithms combine parameterized circuits with classical optimization (Cerezo et al., 2021). Their performance depends on parameter setting (Sud et al., 2024), train-

ability (McClean et al., 2018), and hardware and optimization choices (Blekos et al., 2024). Recent quantum-model benchmarks further motivate strong classical controls and component ablations (Bowles et al., 2024). QIOCE evaluates its counterfactual objective classically on decoded features: its phase layer is not claimed to implement a Hamiltonian whose expectation equals that objective (Luna et al., 2025). The question is whether this QAOA-inspired parameterization supplies useful search candidates, rather than whether a large state space alone implies an advantage.

Visual counterfactuals and adversarial examples. Prediction-changing perturbations also underlie adversarial examples (Goodfellow et al., 2015). Diffusion-based image counterfactuals use generative structure to guide changes (Jeanneret et al., 2024), while latent-diffusion approaches combine visual localization with adversarial guidance (Luu et al., 2025). Our digit experiments illustrate nearby model-level counterfactuals and adversarial-like boundary probes; they are intended as qualitative demonstrations rather than comparisons with these higher-resolution generative methods. Classifier validity, visual similarity, and target-class plausibility remain distinct properties.

3 METHOD

3.1 PROBLEM AND TRAINING REGIME

Let $x = (u, c)$ contain d continuous features u and m categorical features c, and let f be a fixed binary classifier. For target $t \neq f(x)$, we seek an alternative x' that satisfies $f(x') = t$ while minimizing proximity, feature changes and a distributional penalty. The implementation uses weighted penalties during search and verifies validity on the final output; it does not guarantee a valid solution on every attempt.

The resulting directional parameter sets $\theta^{\text{train}}_{0\to1}$ and $\theta^{\text{train}}_{1\to0}$ are then frozen. Here *frozen* means that test examples do not update the saved parameter sets. Per-instance optimization adapts a local copy in two stages. L_1 emphasizes target validity, while L_2 improves counterfactual quality:

$$\theta^{(1)} = \text{Opt}_\theta\left[L_1; x, t, \theta^{\text{train}}_{f(x)\to t}\right], \qquad \theta^{(2)} = \text{Opt}_\theta\left[L_2; x, t, \theta^{(1)}\right]. \tag{1}$$

The objectives L_1 and L_2 are defined in Section 3.3. The decoded candidate $x^{(0)} = I_{\theta^{(2)}}(x)$ initializes refinement $x^{\text{cf}} = R(x^{(0)}; x, t, f)$. Target information enters through the selected directional parameters and the target-class loss, rather than through an additional target qubit. Appendix A gives the complete sequence.

3.2 ENCODING, CIRCUIT AND DECODING

Data encoding determines the function class available to a variational circuit (Schuld et al., 2021; Plewa et al., 2021). Each continuous feature uses one qubit with rotation $R_y(\pi(u_i + 1)/2)$. As compared to the standard fixed-point algorithm, the number of qubits grows here linearly with the number of continuous variables. A categorical feature with K_j possible values uses K_j qubits, initialized by an X gate on the qubit corresponding to its factual category. The total register size is $q = d + \sum_j K_j$.

Each of p layers applies shared $R_z(\gamma_\ell)$ rotations, a forward nearest-neighbor CNOT chain on the continuous register and within each categorical register, then shared $R_x(\beta_\ell)$ rotations on continuous qubits and pairwise XY mixing within categorical registers. The circuit has $2p$ parameters, with $\gamma_\ell \in [0, 2\pi]$ and $\beta_\ell \in [0, \pi]$. Cost and mixer strength multipliers are one. Exact gate order and categorical mixing are given in Appendix A.2.

For output state ρ, continuous decoding uses the reduced state ρ_i:

$$z_i = \text{Tr}(\rho_i Z), \qquad u_i^{(0)} = -z_i, \qquad c_j^{(0)} = \arg\max_{k\in\{0,\ldots,K_j-1\}} \frac{1 - z_{jk}}{2}. \tag{2}$$

Direct access to reduced states is convenient in statevector simulation, but obtaining the full quantum state on hardware would require state tomography, whose cost scales poorly with the number of qubits. To avoid this bottleneck, our decoder relies only on single-qubit marginal Z expectations,

which can all be estimated from the same computational-basis measurement batch:

$$\widehat{u}_i^{(0)} = -\frac{n_i(0) - n_i(1)}{N}, \tag{3}$$

where N is the number of shots, and $n_i(0)$ and $n_i(1)$ denote the marginal measurement counts of qubit i. The same batch therefore provides the statistics required to decode all continuous and categorical features without reconstructing the full quantum state. The categorical argmax returns one allowed category. Although an ideal XY mixer preserves excitation number, the complete circuit also contains CNOT gates; categorical output validity is therefore enforced by decoding, not by a claim that the entire state remains one-hot.

3.3 Two-step objectives and local refinement

Define continuous squared distance $D = \|u' - u\|_2^2$, categorical changes $H = \sum_j \mathbf{1}[c'_j \neq c_j]$, and sparsity $S = \sum_i \mathbf{1}[|u'_i - u_i| > 0.01] + H$. The first step uses normalized versions $\bar{D} = D/(4d)$, $\bar{H} = H/m$, and $\bar{S} = S/(d+m)$, taking an absent feature-type term as zero. For target probability $p_t = f_t(x')$, the implemented validity loss is

$$C_t(x') = \begin{cases} 2 + 4(0.5 - p_t), & p_t < 0.5, \\ (1 - p_t)/2, & p_t \geq 0.5. \end{cases} \tag{4}$$

A Gaussian KDE supplies density guidance $G_t(x') = -\exp(\mathrm{score}_{\mathrm{KDE,t}}(x')) = -\widehat{p}_t(x')$, with bandwidth 0.5. The class-aware variant fits the KDE to target-class training data. The two objectives are

$$L_1 = \bar{D} + \bar{H} + 5C_t + \alpha_1 \bar{S} + \eta_1 G_t + B_1, \tag{5}$$

$$L_2 = D + H + 0.5S + \eta_2 G_t + B_2. \tag{6}$$

Here α_1 controls the sparsity contribution in Step 1, while η_1 and η_2 weight the KDE-based density guidance G_t in Steps 1 and 2, respectively. Their values depend on the implementation branch and are specified in Appendix A.3. The terms B_1 and B_2 collect enabled distributional penalties. Both stages use bounded L-BFGS-B Sen et al. (2022). Step 2 starts from Step 1's parameters and omits the classification loss, so validity must be checked again after quality optimization.

The optional model refiner has two phases. If the initial candidate is invalid, it searches coordinate perturbations and categorical alternatives until it finds a target-class candidate. It then accepts valid moves that reduce continuous squared distance plus categorical changes. This is distinct from the external stochastic refiner used to compare initializers; that refiner and its acceptance rule are specified in Appendix A.4. Feature-range and standard-deviation penalties are soft constraints, and immutable features are not hard-constrained.

4 Experiments

4.1 Training Details and Evaluation

The full-pipeline study covers Banknote, Blood Transfusion, Diabetes, Haberman, Heart Disease, Titanic and Wine. The initialization study covers Moons, Iris, Wine, Titanic and Diabetes. QIOCE's tabular training regime uses a random-forest classifier trained once per run, directional parameter learning on the training set, and test-time adaptation from frozen starting parameters. For the original tabular comparisons, the reported seed set is $\{42, 66, 79\}$; reported means average the seed-level results, and $\pm$ denotes the standard deviation across the three seed-level means. The supplementary diagnostics below are separate single-run studies and do not inherit these uncertainty estimates.

The tabular variational configuration uses five layers and L-BFGS-B, with a 100-iteration limit for Step 1. The two-step implementation exposes a separate Step 2 limit (default 50; the class-aware directional runner sets 100). The initialization comparison uses a shared external refiner with 100 proposals. Appendix C describes preprocessing and separates implementation settings from the distinct qualitative experiments; iteration limits are not treated as measured evaluation counts.

Table 1 compares counterfactual quality on the full evaluated population: all methods generate valid counterfactuals for 200/200 (54/54 for Wine, 92/92 for Haberman, 133/133 for Diabetes, 91/91 for

Heart disease) attempted instances. We report L2 distance, sparsity, and nearest-neighbor plausibility. The plausibility metric is distinct from the KDE guidance used during QIOCE optimization. Table 2 instead retains failures and reports validity together with mean distance over all attempts, so the two quantities must be interpreted jointly. Complete metric definitions are given in Appendix C.2.

4.2 Main Results

Quality of tabular counterfactuals. In Table 1, all methods generate valid counterfactuals for all attempted instances, so the comparison reflects unconditional counterfactual quality on the full evaluated population. QIOCE achieves the lowest mean L2 distance on five of seven datasets: Blood Transfusion, Diabetes, Heart Disease, Titanic, and Wine. Wachter has the lowest distance on Banknote, and DiCE on Haberman. QIOCE also achieves the lowest plausibility distance on all seven datasets.

Image counterfactuals show visible changes in a model's decision. Figure 2 shows factual digits, prediction-changing alternatives, and their pixel-change maps. For each digit pair, the classifier operates in PCA space, while the displayed images are reconstructed to 8×8 pixels. Several counterfactuals remain visually close to the corresponding factual examples while changing the classifier's prediction to the target class. This illustrates an important distinction: a counterfactual only needs to change the model's decision; it does not necessarily need to look like a prototypical example of the target class. The visualizations make this behavior easy to inspect while keeping model prediction and human interpretation conceptually separate.

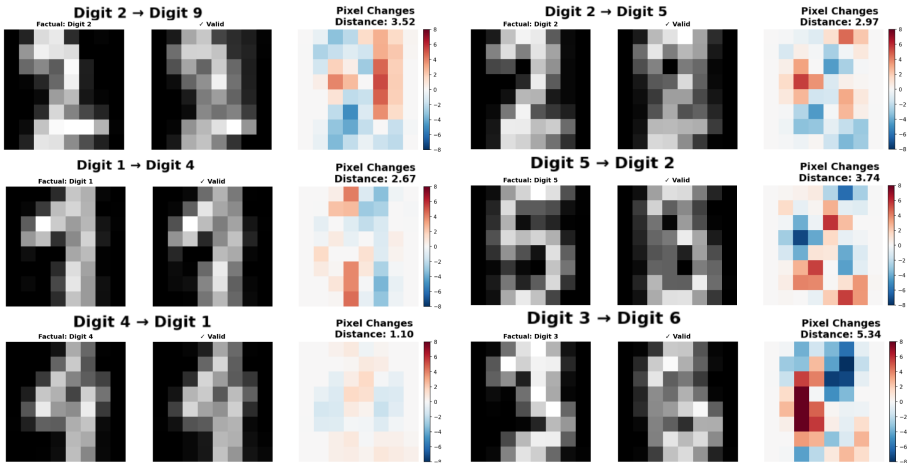


Figure 2: Digits counterfactuals with class-aware guidance. Each group shows a reconstructed factual, a target-valid alternative, and a pixel-change map. Digit labels refer to the binary task's original digit identities. Distances are in the optimization representation; colors show reconstructed pixel differences. Visual resemblance can persist after the classifier changes its prediction.

Figure 3 shows a second use of the same framework, with weaker distributional guidance. In this setting, the objective is primarily to find a nearby input that changes the classifier's prediction, rather than to produce an example that strongly resembles the target-class data distribution. The resulting samples therefore act as adversarial-like probes of the decision boundary: they reveal how small changes to a factual input can move it across the classifier's boundary. They are still counterfactuals with respect to the model, but here the emphasis is on boundary sensitivity rather than target-class plausibility. The examples use different factual–target pairs and are intended as qualitative illustrations, not as a controlled comparison of density guidance or as an adversarial-attack benchmark.

4.3 Ablation Study

Replacing the initializer. Table 2 compares variational initialization with Random, KNN, Boundary, SMOTE-like interpolation, simulated annealing and a genetic algorithm. All feed the same

Dataset	Method	L2 Distance ↓	Sparsity ↓	Plausibility ↓
Banknote	QIOCE	0.123 (±0.041)	4.00 (±0.00)	**0.111** (±0.005)
	Growing Spheres	0.063 (±0.025)	4.00 (±0.00)	0.175 (±0.002)
	Wachter	**0.022** (±0.007)	**1.00** (±0.00)	0.126 (±0.006)
	DiCE	0.112 (±0.037)	3.00 (±0.00)	0.341 (±0.021)
	TABCF	0.091 (±0.029)	3.00 (±0.00)	0.143 (±0.009)
	Tab. Diffusion	0.134 (±0.033)	3.50 (±0.20)	0.198 (±0.013)
Blood transfusion	QIOCE	**0.311** (±0.012)	4.00 (±0.00)	**0.089** (±0.004)
	Growing Spheres	0.836 (±0.057)	4.00 (±0.00)	0.279 (±0.025)
	Wachter	0.615 (±0.072)	2.80 (±0.00)	0.297 (±0.039)
	DiCE	0.382 (±0.018)	**2.20** (±0.10)	0.251 (±0.024)
	TABCF	0.348 (±0.021)	2.60 (±0.10)	0.172 (±0.017)
	Tab. Diffusion	0.412 (±0.034)	3.00 (±0.00)	0.213 (±0.022)
Diabetes	QIOCE	**0.487** (±0.021)	**9.10** (±0.10)	**0.315** (±0.024)
	Growing Spheres	2.295 (±0.371)	10.00 (±0.00)	1.827 (±0.081)
	Wachter	2.069 (±0.226)	9.20 (±0.30)	1.650 (±0.083)
	DiCE	1.740 (±0.087)	9.50 (±0.10)	1.102 (±0.171)
	TABCF	0.731 (±0.059)	9.40 (±0.20)	0.618 (±0.052)
	Tab. Diffusion	0.994 (±0.083)	9.60 (±0.10)	0.847 (±0.074)
Haberman	QIOCE	0.464 (±0.009)	**2.00** (±0.00)	**0.090** (±0.011)
	Growing Spheres	1.213 (±0.066)	3.00 (±0.00)	0.346 (±0.029)
	Wachter	0.750 (±0.012)	3.00 (±0.00)	0.350 (±0.019)
	DiCE	**0.384** (±0.010)	**2.00** (±0.00)	0.322 (±0.015)
	TABCF	0.512 (±0.018)	2.50 (±0.00)	0.241 (±0.018)
	Tab. Diffusion	0.631 (±0.027)	2.80 (±0.10)	0.279 (±0.021)
Heart disease	QIOCE	**0.401** (±0.011)	**11.00** (±0.00)	**0.119** (±0.011)
	Growing Spheres	0.822 (±0.039)	13.00 (±0.00)	0.121 (±0.021)
	Wachter	0.450 (±0.005)	13.00 (±0.00)	0.164 (±0.025)
	DiCE	0.421 (±0.008)	11.20 (±0.10)	0.168 (±0.027)
	TABCF	0.434 (±0.014)	11.80 (±0.10)	0.138 (±0.016)
	Tab. Diffusion	0.468 (±0.019)	12.00 (±0.00)	0.147 (±0.019)
Titanic	QIOCE	**0.175** (±0.027)	3.00 (±0.00)	**0.074** (±0.041)
	Growing Spheres	0.489 (±0.105)	5.00 (±0.00)	0.373 (±0.072)
	Wachter	0.254 (±0.054)	2.60 (±0.10)	0.328 (±0.059)
	DiCE	0.221 (±0.062)	**2.40** (±0.20)	0.317 (±0.033)
	TABCF	0.203 (±0.038)	2.80 (±0.10)	0.189 (±0.027)
	Tab. Diffusion	0.247 (±0.049)	3.00 (±0.00)	0.241 (±0.031)
Wine	QIOCE	**0.602** (±0.088)	13.00 (±0.00)	**0.474** (±0.110)
	Growing Spheres	2.599 (±0.345)	13.00 (±0.00)	2.402 (±0.188)
	Wachter	2.648 (±0.191)	13.00 (±0.00)	2.141 (±0.200)
	DiCE	1.214 (±0.080)	13.00 (±0.00)	0.987 (±0.039)
	TABCF	0.841 (±0.071)	13.00 (±0.00)	0.712 (±0.058)
	Tab. Diffusion	1.073 (±0.094)	13.00 (±0.00)	0.831 (±0.066)

Table 1: Quality comparison on the full tabular evaluation population. All methods generate valid counterfactuals for all attempted instances, so the reported L2 distance, sparsity, and plausibility values are unconditional full-population summaries. Entries are mean ± standard deviation across three seed-level means (42, 66, 79). Lower values are preferred.

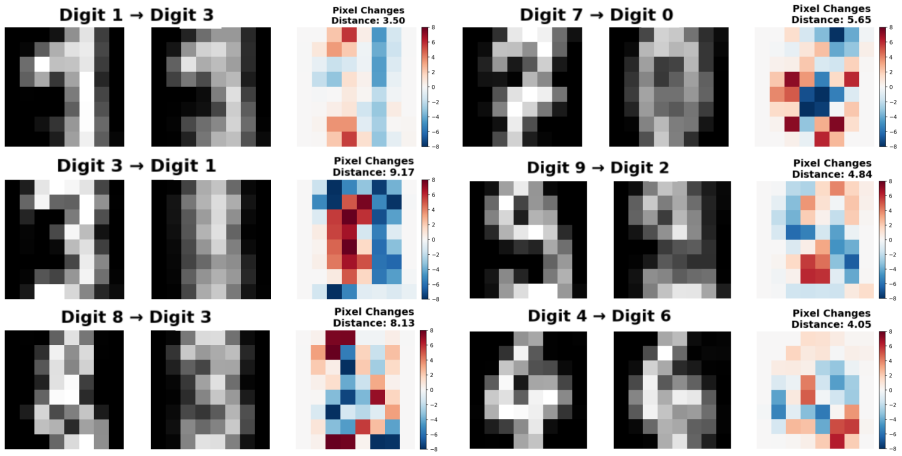


Figure 3: Adversarial-like Digits counterfactuals under weaker distributional guidance. The classifier's target prediction can be reached while retaining visual features of the factual input. Change maps highlight the resulting perturbations; reported distances are measured in feature space and should not be interpreted as pixel-space attack budgets.

external refinement rule. QIOCE has the highest or tied-highest validity on four of five datasets. On Titanic, Boundary and simulated annealing achieve 85% compared with QIOCE's 75%. On Wine, KNN matches QIOCE's 90% validity with a smaller mean all-output distance (2.492 versus 2.710). These outcomes establish a dataset-dependent validity–distance trade-off, not dominance on both axes.

Component ablation. On Wine, we additionally compare QIOCE with randomized variational initialization (QIOCE$_{R}andInit$), $directoptimization, andgeneticsearchunderasharedrefinementbudget(Appendix\ E.3$

Method	Metric	Moons	Iris	Wine	Titanic	Diabetes
QIOCE	Validity (%)	**95.0**	**100.0**	**90.0**	75.0	**100.0**
	Distance	0.782	1.663	2.710	1.983	2.096
Random	Validity (%)	15.0	5.0	5.0	55.0	45.0
	Distance	0.511	0.991	1.774	0.917	1.190
KNN	Validity (%)	75.0	**100.0**	**90.0**	60.0	85.0
	Distance	0.477	2.166	2.492	0.800	0.989
Boundary	Validity (%)	70.0	**100.0**	87.0	**85.0**	95.0
	Distance	0.445	1.695	2.087	0.496	0.847
SMOTE	Validity (%)	5.0	50.0	40.0	35.0	70.0
	Distance	0.309	1.481	1.630	0.493	0.830
Sim. Annealing	Validity (%)	85.0	30.0	45.0	**85.0**	70.0
	Distance	0.686	1.708	1.630	1.577	0.787
Genetic Alg.	Validity (%)	85.0	5.0	25.0	75.0	65.0
	Distance	0.390	0.256	0.702	0.416	0.329

Table 2: Initialization comparison after shared refinement. Validity and distance are computed over all attempts, including failures; distance is therefore not conditional on successful counterfactual generation. Bold validity entries indicate the highest or tied-highest value within each dataset.

What the baselines compute. Random adds Gaussian noise to the factual. KNN returns an inverse-distance-weighted centroid of five target-labeled training neighbors. Boundary bisects the segment toward the nearest target-labeled neighbor and steps beyond the estimated boundary. SMOTE-like initialization interpolates toward a randomly selected target neighbor. Simulated an-

nealing trades target probability against distance; the genetic algorithm searches a population with a probability–distance fitness. Appendix B gives coefficients, stopping rules and computational costs. These strategies share refinement, but do not receive equal initialization effort; the experiment assesses their configured pipelines rather than matched-budget efficiency.

Effect of refinement. Figure 4 reports a separate Moons experiment with both stages visible. Quantum initialization moves from 80% to 86% validity, with mean distance decreasing from 0.878 to 0.567. KNN and Boundary move from 48% to 76%, with final distances of 0.516 and 0.488; the genetic initializer reaches 78% validity at distance 0.379. This separates the starting candidate's behavior from the improvement added by local search. These figure results are a different experiment from the multi-dataset table and are not pooled with it.

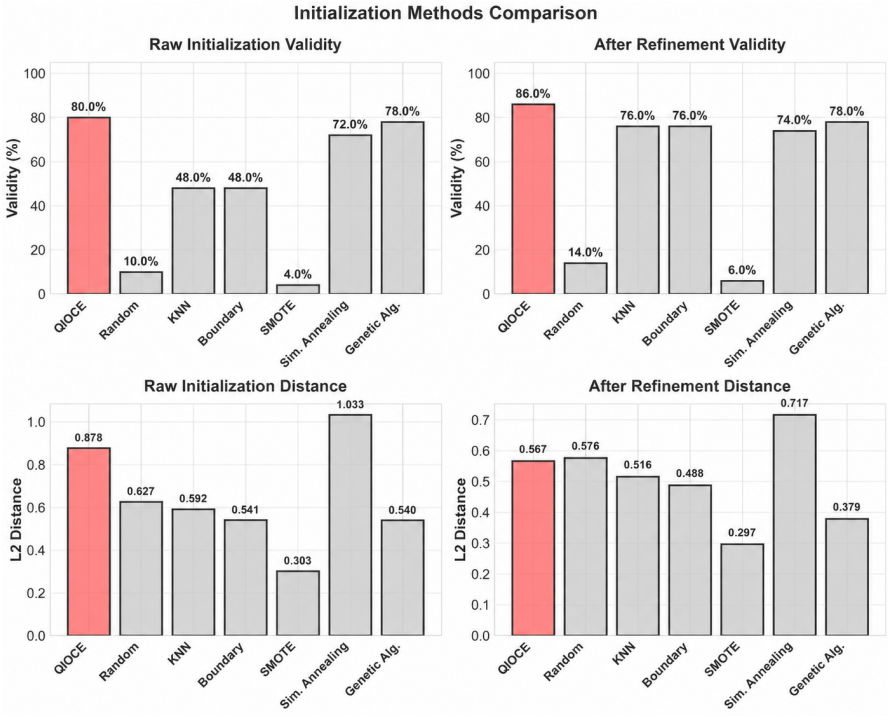


Figure 4: Moons initialization experiment before and after shared refinement. The four panels expose both validity and mean distance at each stage. The best final validity need not coincide with the largest refinement gain or the smallest distance.

Measurement and hardware feasibility. Equation 3 permits decoding directly from measurement counts without reconstructing the full quantum state. We executed QIOCE on real quantum hardware using circuits ranging from 2 to 150 qubits. On `ibm_fez`, runs with 8192 shots produced target-class counterfactual candidates directly from hardware measurements. Appendix D presents the measurement-based decoding procedure and its behavior under noise, and distinguishes the expectation and inverse-angle decoders. These experiments demonstrate that the proposed decoding and generation pipeline can be executed on current quantum hardware; they do not establish a computational quantum advantage or guarantee that the multi-dataset simulation results are preserved under hardware noise.

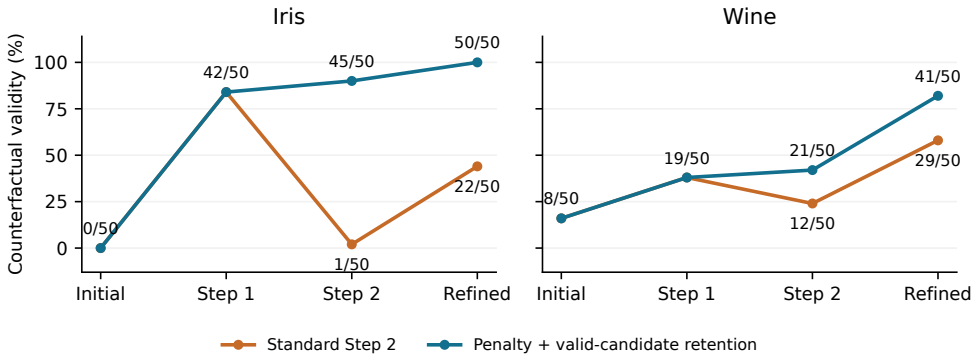


Figure 5: Iris and Wine panels of the four-dataset stage study. Counts are valid counterfactuals out of 50. The retention variant adds the same invalid-region penalty on these two datasets and caches valid candidates; its additional optimization cost is not controlled. Full results and costs are in Appendix E.

5 Limitations and Conclusion

We introduced QIOCE, a hybrid quantum–classical framework that combines direction-specific variational initialization with instance-level adaptation and classical refinement for counterfactual search. Across the tabular benchmarks, QIOCE achieves the lowest valid-counterfactual distance on five of seven datasets and the highest or tied-highest validity on four of five datasets in the shared-refinement study. Stage-wise diagnostics show that refinement can remove validity, while retaining valid candidates mitigates this failure mode. The same framework also produces visual counterfactuals and adversarial-like boundary probes, and supports measurement-based execution on real quantum hardware.

These results support variational quantum initialization as a candidate-generation mechanism, but do not establish computational quantum advantage. The main limitations are unmatched computational budgets, limited statistical evidence, soft rather than hard feasibility constraints, qualitative image evaluation, and the absence of a systematic noisy-hardware robustness study.

Future work should focus on matched-budget comparisons, stronger constraint handling, and repeated hardware evaluation under noise.

Reproducibility Statement

The method description distinguishes offline parameter transfer from test-time adaptation. The appendices specify circuit operations, loss functions, refinement rules, baseline configurations, evaluation metrics, and the protocols used for the supplementary studies. The reported seed set is 42, 66, and 79, and Table 1 reports standard deviations across seed-level means. The full implementation and experiment scripts will be released publicly after the review process.

AI Use Statement

An AI assistant supported manuscript editing, organization, implementation inspection and figure layout. Experimental results were supplied by the authors; no new benchmark or hardware results were generated for this revision.

Ethics Statement

Counterfactuals are model-level prediction changes and should not be interpreted as feasible interventions or recommendations without explicit actionability and causal constraints.

REFERENCES

Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 1068:1–66, 2024. URL https://doi.org/10.1016/j.physrep.2024.03.002.

Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059*, 2024. URL https://arxiv.org/abs/2403.07059.

M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 2021. URL https://doi.org/10.1038/s42254-021-00348-9.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature – PPSN XVI*, 2020. URL https://arxiv.org/abs/2004.11165.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014. URL https://arxiv.org/abs/1411.4028.

Oleksii Furman, Ulvi Movsum-zada, Patryk Marszalek, Maciej Zięba, and Marek Śmieja. DiCoFlex: Model-agnostic diverse counterfactuals with flexible control. *arXiv preprint arXiv:2505.23700*, 2025. URL https://arxiv.org/abs/2505.23700.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6572.

Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 38:2770–2824, 2024. URL https://doi.org/10.1007/s10618-022-00831-6.

Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019. URL https://doi.org/10.3390/a12020034.

Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. Diffusion models for counterfactual explanations. *Computer Vision and Image Understanding*, 249:104207, 2024. URL https://doi.org/10.1016/j.cviu.2024.104207.

Junqi Jiang, Jianglin Lan, Francesco Leofante, Antonio Rago, and Francesca Toni. Provably robust and plausible counterfactual explanations for neural networks via robust optimisation. In *Proceedings of the 15th Asian Conference on Machine Learning*, volume 222 of *Proceedings of Machine Learning Research*, pp. 582–597, 2024a. URL https://proceedings.mlr.press/v222/jiang24a.html.

Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Robust counterfactual explanations in machine learning: A survey. *arXiv preprint arXiv:2402.01928*, 2024b. URL https://arxiv.org/abs/2402.01928.

Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362, 2021. URL https://doi.org/10.1145/3442188.3445899.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017. URL https://arxiv.org/abs/1712.08443.

Miguel Luna, Vedant Patare, Gamzepelin Aksoy, and Grégoire Cattan. Implementation of a quantum approximate optimization algorithm for continuous variables with qiskit. HAL preprint, 2025. HAL: hal-05295918.

Tung Luu, Nam Le, Duc Le, and Bac Le. From visual explanations to counterfactual explanations with latent diffusion. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 420–429, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Luu_From_Visual_Explanations_to_Counterfactual_Explanations_with_Latent_Diffusion_WACV_2025_paper.html.

Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9:4812, 2018. URL https://doi.org/10.1038/s41467-018-07090-4.

Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020. doi: 10.1145/3351095.3372850. URL https://arxiv.org/abs/1905.07697.

Emmanouil Panagiotou, Manuel Heurich, Tim Landgraf, and Eirini Ntoutsi. TABCF: Counterfactual explanations for tabular data using a transformer-based VAE. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 274–282, 2024. doi: 10.1145/3677052.3698673. URL https://arxiv.org/abs/2410.10463.

Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. CARLA: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint arXiv:2108.00783*, 2021. URL https://arxiv.org/abs/2108.00783.

Joanna Plewa, Jakub Sieńko, and Katarzyna Rycerz. Variational algorithms for workflow scheduling problem in gate-based quantum devices. *Computing and Informatics*, 40(4):897–929, 2021.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. *arXiv preprint arXiv:1909.09369*, 2019. URL https://arxiv.org/abs/1909.09369.

Maximilian Schleich, Zixuan Geng, Yihong Zhang, and Dan Suciu. GeCo: Quality counterfactual explanations in real time. *arXiv preprint arXiv:2101.01292*, 2021. URL https://arxiv.org/abs/2101.01292.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. The effect of data encoding on the expressive power of variational quantum machine learning models. *Physical Review A*, 103:032430, 2021. URL https://doi.org/10.1103/PhysRevA.103.032430.

Pinaki Sen, Amandeep Singh Bhatia, Kamalpreet Singh Bhangu, and Ahmed Elbeltagi. Variational Quantum Classifiers Through the Lens of the Hessian. *PLOS ONE*, 17(1):e0262346, January 2022. ISSN 1932-6203. doi: 10.1371/journal.pone.0262346. URL http://arxiv.org/abs/2105.10162. arXiv:2105.10162 [quant-ph].

James Sud, Stuart Hadfield, Eleanor Rieffel, Norm Tubman, and Tad Hogg. Parameter-setting heuristic for the quantum alternating operator ansatz. *Physical Review Research*, 6:023171, 2024. URL https://doi.org/10.1103/PhysRevResearch.6.023171.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019. doi: 10.1145/3287560.3287566. URL https://arxiv.org/abs/1809.06514.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019. URL https://arxiv.org/abs/1907.02584.

Germain Vivier-Ardisson, Alexandre Forel, Axel Parmentier, and Thibaut Vidal. CF-OPT: Counterfactual explanations for structured prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49558–49579, 2024. URL https://proceedings.mlr.press/v235/vivier-ardisson24a.html.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 2018. URL https://arxiv.org/abs/1711.00399.

Wei Zhang, Brian Barr, and John Paisley. Tabular diffusion counterfactual explanations. *Frontiers in Artificial Intelligence*, 9:1743495, 2026. URL https://doi.org/10.3389/frai.2026.1743495.

A Algorithm and Implementation Details

A.1 Training and inference

The directional two-step implementation provides standard and class-aware variants. The latter adds target-class KDE guidance. The separate binary/domain-wall encoding implementation is not interchangeable with the one-hot circuit described here.

Stage	Procedure
Training	Fit preprocessing and the classifier on training data. Fit KDEs and enabled distribution statistics. For each direction $0 \to 1$ and $1 \to 0$, initialize parameters and process training factuals sequentially. Run Steps 1 and 2, transferring the optimized parameters to the next training factual in that direction. Store the two resulting parameter sets and freeze their persistent values.
Inference	Predict $a = f(x)$ and choose $t \neq a$. Copy $\theta^{\text{train}}_{a\to t}$; encode x; optimize L_1 with bounded L-BFGS-B; initialize Step 2 from the Step 1 result and optimize L_2; decode the resulting circuit; optionally refine; verify $f(x^{\text{cf}}) = t$. Return the candidate with its validity flag. Discard test-time parameter updates before processing the next factual.
Comparison	Replace the entire quantum initializer with a classical initializer. Disable internal model refinement when applying the shared external refiner. Evaluate the same final validity and distance definitions for each method.

Table 3: QIOCE training, inference and initialization-comparison protocol. The frozen quantities are shared starting parameters, not the local variables optimized for each test instance.

The two optimization steps both operate on circuit parameters. Classical feature-space refinement is a subsequent operation. This distinguishes three effects: transfer from training instances, adaptation of the circuit to the current instance, and local movement of the decoded candidate. Because Step 2 drops classification loss, its output is not assumed valid merely because Step 1 reached the target.

A.2 Exact circuit structure

Registers are ordered with categorical qubits followed by continuous qubits. All start in $|0\rangle$. For categorical register j, apply X to its factual-category qubit; for continuous coordinate i, apply $R_y(\pi(u_i + 1)/2)$. Each layer uses the following order:

1. Apply $R_z(\gamma_\ell)$ to every qubit.
2. Apply CNOTs successively along the continuous chain, from the first continuous qubit to its successor and onward. Apply the same forward-chain pattern separately inside each categorical register. The one-hot implementation has no closing ring edge and no cross-register CNOTs.
3. Apply $R_x(\beta_\ell)$ to each continuous qubit. For each categorical register and each pair $r < s$, apply $\exp(-i\beta_\ell X_r X_s/2)$ followed by $\exp(-i\beta_\ell Y_r Y_s/2)$, in lexicographic pair order.

The XX and YY rotations are decomposed into basis changes, CNOT–R_z–CNOT sequences, and inverse basis changes. Within a pair, the XX and YY terms commute; operations on overlapping pairs are applied sequentially. The R_z layer is a trainable phase layer, while classifier and KDE information enters the classical objective.

The ideal XY operator preserves excitation number, but a categorical-chain CNOT need not: for example, a CNOT can map $|10\rangle$ to $|11\rangle$. The final argmax over category-qubit excitation probabilities resolves the output to a single allowed category even in this case. This guarantees a categorical assignment at the decoder, not one-hot support throughout the quantum state.

For exact simulation, ρ_i in Equation 2 is computed by tracing out every qubit except i. For sampled execution, marginal bit counts supply the same expectation decoder. Qiskit count strings place qubit zero at the rightmost bit. The inverse-angle decoder $2\arccos(z_i)/\pi - 1$ is a separate map; it agrees with the input coordinate for a bare rotation encoding on $[-1, 1]$, but is not identical to $-z_i$. The expectation decoder is a bounded candidate parameterization, not an exact inverse of the input-angle map.

A.3 Loss coefficients and optional penalties

Table 4 records the implemented loss branches. A term is zero when the relevant KDE is absent. The class-aware branch uses target-class KDE density; the standard branch uses a pooled training KDE. All KDEs use a Gaussian kernel with bandwidth 0.5 in the model's feature coordinates.

Implementation branch	Step 1 α_1	Step 1 η_1	Step 2 sparsity	Step 2 η_2
Standard weighted objective	0.2	0	0.5	0.3
Standard, Pareto tracking enabled	0.3	0.3	0.5	0.3
Class-aware weighted objective	0.2	0.8	0.5	0.8
Class-aware, Pareto tracking enabled	0.3	0.8	0.5	0.8

Table 4: Loss coefficients in the two implementation variants. Step 1 weights classification by 5.0 and uses normalized distance and sparsity; Step 2 uses raw squared distance and feature-change counts and has no classification term. Pareto tracking still uses a scalar weighted objective for optimization.

Enabled distribution penalties add the following terms, with $[a]_+ = \max(a, 0)$:

$$B_{\text{std}}(u') = w_{\text{std}} \sum_i \left[\frac{|u'_i - \mu_i|}{\sigma_i} - n_{\text{std}} \right]_+^2, \tag{7}$$

$$B_{\text{range}}(u') = w_{\text{range}} \sum_i \left([a_i - u'_i]_+^2 + [u'_i - b_i]_+^2 \right), \tag{8}$$

$$B_{\text{density}}(x') = -w_{\text{density}} \log \left(\widehat{p}_{\text{pooled}}(x') + 10^{-10} \right). \tag{9}$$

The defaults are $w_{\text{std}} = 2$, $n_{\text{std}} = 2$, $w_{\text{range}} = 1$ and $w_{\text{density}} = 1.5$. Bounds a_i, b_i and statistics μ_i, σ_i come from the fitted training statistics. The density penalty is separate from the negative-density term ηG_t: the class-aware implementation can combine target-class density guidance with a pooled-density penalty. Step 1 also penalizes a selected categorical value whose training frequency is below 0.05, using $w_{\text{std}}[0.05 - \Pr(c'_j)]_+^2$ when the corresponding flag is enabled. Step 2 does not add this categorical-frequency term.

These penalties are not projections onto a feasible set. They do not enforce immutability or causal constraints, and the local refiner does not automatically preserve every soft penalty value. Neither a density value nor a valid categorical assignment certifies actionability.

A.4 Two refinement implementations

Optional model refiner. The first phase is entered only if the decoded candidate is invalid. It tests $\pm h$ changes to each continuous coordinate and each alternative categorical value. It accepts the first target-valid candidate even if its distance is larger. Initially $h = 0.1$; after each 20 unsuccessful sweeps, $h \leftarrow \min(1.5h, 0.5)$. The default limit is 100 sweeps. If no valid candidate is found, it returns the initial candidate and the caller evaluates its validity.

The second phase starts from a valid candidate and uses continuous coordinate proposals of size 0.05, plus attempts to restore changed categorical values to their factual values. It accepts target-valid proposals only when $D + H$ decreases. The implementation has a separate stopping variable initialized to 0.1, halved on sweeps without improvement and stopped below 10^{-6}; this variable does not change the fixed 0.05 proposal size. The maximum is 100 sweeps. The proposal size remains fixed throughout the second phase.

Shared external refiner for initialization comparisons. Let b be the initialized candidate and $r = \|b - x\|_2$. Starting step size is $h = 0.05$ if b is target-valid and 0.1 otherwise. At each of 100 iterations, propose $v = b + \epsilon$ with $\epsilon \sim \mathcal{N}(0, h^2 I)$. Accept only if $f(v) = t$ and $\|v - x\|_2 < r$; on acceptance set $b \leftarrow v$, update r, and set $h \leftarrow \max(0.02, 0.95h)$. For invalid proposals after a valid solution has been found, use $h \leftarrow \max(0.02, 0.98h)$. While no valid solution has been found, invalid proposals at iterations divisible by 20 increase the step to $\min(1.2h, 0.5)$. Return b and check validity again.

The distance threshold initially equals the initializer's distance even when that initializer is invalid. Consequently, this refiner characterizes a strict distance-improving search policy. Sharing the policy makes the initializer comparison explicit; it does not establish that the same ranking holds for a validity-first acceptance policy.

B Baseline Definitions and Computational Effort

Full-pipeline baselines. The bundled CARLA implementations provide the three counterfactual baselines in Table 1. Growing Spheres samples expanding shells around a factual and selects a prediction-changing candidate; its library defaults use 1000 samples per shell, L2 distance and shell increments of 0.2. Wachter optimizes a classification-and-distance objective; the wrapper defaults use binary cross-entropy, L1 feature cost, learning rate 0.01, distance weight 0.01, and an iteration limit of 1000 with a 0.5-minute time limit. The runner disables binary categorical encoding for this wrapper. The DiCE wrapper uses its random-search mode, requests one counterfactual, and sets post-hoc sparsity to 0.1. These describe the available wrapper configurations, not an assertion that their search costs are equal. They are complete generators, distinct from the starting-point heuristics below.

Generative baselines. We additionally compare against TABCF (Panagiotou et al., 2024), a transformer-VAE-based method for mixed tabular data, and Tabular Diffusion (Zhang et al., 2026), which generates counterfactuals using a diffusion-based tabular model. Both methods are evaluated using the same train/test splits and the same final validity, distance, sparsity, and plausibility metrics as the other methods in Table 1. Across the seven datasets, QIOCE achieves a lower mean ℓ_2 distance than TABCF on six datasets and than Tabular Diffusion on all datasets. For example, on Diabetes the mean distance is 0.487 for QIOCE, compared with 0.731 for TABCF and 0.994 for Tabular Diffusion, while on Wine the corresponding values are 0.602, 0.841, and 1.073. The same pattern is visible in the nearest-neighbor plausibility metric: on Diabetes the values are 0.315, 0.618, and 0.847, respectively, and on Wine 0.474, 0.712, and 0.831.

All definitions below follow the initializer implementations. Let x be the factual and $f_t(v)$ the classifier's target probability. Target-neighbor selection uses training labels, not a fresh check that the classifier assigns every neighbor to the target class.

Random. Return $x + \epsilon$, $\epsilon \sim \mathcal{N}(0, 0.5^2 I)$. One candidate is generated. There is no internal classifier call; the harness evaluates its validity afterward.

KNN. Select the five closest target-labeled training examples (or all available examples if fewer than five), using Euclidean distance. Return their centroid weighted by $(d_k + 10^{-10})^{-1}$ and normalized to sum to one. The neighbor search uses no classifier calls internally; its distance computations remain a computational cost.

Boundary. Find the nearest target-labeled training point v. Bisect the segment $x + \alpha(v - x)$ with $\alpha \in [0, 1]$, shrinking the upper endpoint when the classifier predicts the target. Stop after 50 iterations or when the interval width is below 10^{-4}. Return the upper endpoint with an additional $0.1(v - x)$ displacement toward v. With the interval-width rule, at most 14 bisections are needed in exact arithmetic; 50 is the configured ceiling, not the realized number of calls.

SMOTE-like interpolation. Select a uniformly random member of the five nearest target-labeled training neighbors. Return $x + \alpha(v - x)$ with $\alpha \sim \mathcal{U}(0.3, 0.7)$. This is a candidate-initialization heuristic, not a claim to reproduce the full SMOTE training-data resampling algorithm. It uses no internal classifier query.

Simulated annealing. Minimize $E(v) = -f_t(v) + 0.1\|v - x\|_2$. Start at x with temperature $T = 1$. For 100 iterations, propose $v' = v + \mathcal{N}(0, (0.5T)^2 I)$; accept an improving move or accept with probability $\exp(-(E(v') - E(v))/T)$. Retain the lowest-energy point encountered and cool with $T \leftarrow 0.95T$. This requires 101 probability evaluations, including the initial point.

Genetic algorithm. Maximize $F(v) = f_t(v) - 0.5\|v - x\|_2$. Initialize 20 candidates with Gaussian perturbations of standard deviation 0.5. Run 50 generations using tournament selection of size three, pairwise convex crossover with probability 0.7, and individual mutation with probability 0.1 and Gaussian scale 0.2. Return the highest-fitness final individual. The default implementation evaluates $20 \times 50 + 20 = 1020$ individuals, including the final population.

Gradient warm start (implemented auxiliary baseline). Use central finite differences with perturbation 10^{-4} to ascend target probability. Normalize a nonzero gradient, start with step 0.05 and decrease it linearly to approximately 0.025 over 100 steps. Stop early if the target prediction has probability above 0.7. Each step uses $2d$ probability calls for the finite differences, one to assess the updated point, and one label call; there is also an initial probability call. This method is described for completeness but is not a row in Table 2.

Quantum and shared refinement costs. A quantum objective evaluation constructs and evaluates a circuit, decodes its marginals, and evaluates the relevant loss. Step 1 queries target probability; Step 2 omits that query from its loss. L-BFGS-B uses numerical differentiation and line search, so a 100-plus-50 iteration configuration does not imply 150 circuit evaluations or 150 classifier calls. With $2p$ parameters, numerical-gradient and line-search work can require several objective evaluations per iteration. The shared external refiner adds 100 proposal label queries plus initial and final checks, independently of initializer-specific work. These counts exclude harness diagnostics, training and neighbor-index construction. We do not infer a runtime advantage or an equal-query comparison from the common refinement limit.

C Experimental Protocol and Metrics

C.1 Preprocessing and study boundaries

The tabular loader implementations for Iris, Wine, Diabetes, Banknote, Blood Transfusion, Haberman, Heart Disease and Titanic use a stratified 70/30 train/test split, training-fitted standardization, and division by the maximum absolute standardized training value. Their random-forest configuration uses 100 estimators and maximum depth five. Iris is converted to Setosa versus the other classes; Wine to class zero versus the other classes; the scikit-learn Diabetes regression target is thresholded at its median to define a binary task. The Diabetes representation contains ten features.

The repository also contains a legacy Moons loader that standardizes before an 80/20 split and separate experiment runners with their own preprocessing. Thus, preprocessing should not be inferred from a dataset name alone. The separate Moons stage plots are not merged with the multi-dataset table, and implementation defaults are not evidence that every saved experiment used them. In particular, the class-aware directional runner sets five layers, a 100-iteration limit for each quantum stage, and up to 50 training factuals per direction; the base model's Step 2 default is 50 iterations. These are distinct configurations.

QIOCE's primary tabular protocol uses the reported random-forest classifier and seeds 42, 66 and 79. Model constructors fall back to logistic regression when no classifier is supplied; the image visualization path uses that fallback. A constructor default therefore does not override the classifier specified for an experiment. The seed-level aggregation is

$$\bar{g} = \frac{1}{3} \sum_{s \in \{42,66,79\}} \bar{g}_s, \tag{10}$$

where $\bar{g}_s$ is the mean over that run's evaluation population. The printed $\pm$ values are standard deviations of the three $\bar{g}_s$ values. Table 1 uses the full evaluated population, since all methods produce valid counterfactuals for all attempted instances in that setting (200/200 per datasets where it was possible to have 200 instances and 54/54 for Wine, 92/92 for Haberman, 133/133 for Diabetes, 91/91 for Heart disease - due to limited amount of samples). Table 2 instead retains failures when averaging distances, because its purpose is to compare initializer performance under shared refinement

C.2 Evaluation Metrics

The optimization objectives and the reported evaluation metrics use related but distinct definitions. Optimization uses squared Euclidean distance, whereas evaluation reports the ordinary Euclidean distance

$$D_{\text{eval}}(x, x') = \|u' - u\|_2. \tag{11}$$

For evaluation, sparsity is defined as the number of continuous and categorical features that change between the factual and counterfactual. To avoid counting negligible numerical perturbations as feature changes, we use the same continuous-feature threshold as in the optimization objective:

$$S_{\text{eval}}(x, x') = \sum_i \mathbf{1}[|u'_i - u_i| > 0.01] + \sum_j \mathbf{1}\left[c'_j \neq c_j\right]. \tag{12}$$

Lower S_{eval} therefore indicates that fewer features undergo non-negligible changes.

Counterfactual validity is defined with respect to the fixed classifier and the desired target class:

$$V(x') = \mathbf{1}[f(x') = t]. \tag{13}$$

For a set of N attempted counterfactuals, reported validity is

$$\text{Validity} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}[f(x'_n) = t_n]. \tag{14}$$

Plausibility is evaluated separately from the distributional guidance used during generation. The negative KDE density G_t is an optimization term, whereas an output is evaluated using distances to a reference training set $\mathcal{T}$:

$$P_{\text{NN}}(x') = \min_{z \in \mathcal{T}} \|x' - z\|_2, \tag{15}$$

$$P_{\text{class}}(x') = \min_{z \in \mathcal{T}: f(z) = f(x')} \|x' - z\|_2. \tag{16}$$

Both are lower-is-better distances. The comparison evaluator reports P_{NN} as plausibility and P_{class} as manifold distance. The latter uses the classifier prediction on the training examples. By contrast, a KDE evaluation $\hat{p}_t(x')$ is a higher-is-better density and is distinct from either distance above.

C.3 Image representation and interpretation

The image workflow uses the 8×8 scikit-learn Digits dataset. Each task selects two digit classes and maps them to labels zero and one. It makes a stratified 70/30 split, fits pixel standardization on training images, and then fits PCA on the standardized training data. Counterfactual search and classifier evaluation operate in PCA coordinates. Visualization applies inverse PCA and inverse standardization, reshapes to 8×8, and clips to the pixel range $[0, 16]$. Both the factual and counterfactual displays are reconstructed through this path; the factual display is not necessarily the untouched original image.

The displayed distance is $\|x'_{\text{PCA}} - x_{\text{PCA}}\|_2$. The color map instead shows the difference between reconstructed images. These quantities have different units and should not be interpreted as an ℓ_p pixel threat budget. PCA dimensions and digit pairs are configurable across runs; the two image grids are selected qualitative examples, not an estimate of average image performance. No human study, perceptual similarity score or paired density on/off effect is inferred from the grids.

D Measurement-Based Decoding and Hardware

The continuous expectation decoder requires only marginal Z statistics, all estimable from the same computational-basis shot batch. In exact simulation, evaluating partial traces is often described in the plotting scripts as "state tomography," although an exact statevector calculation is not an experimental tomography protocol.

Figure 6 presents the retained two-panel diagnostic. The left panel shows original and decoded values; the right panel shows the noise-related error comparison in the same setup, as specified for

the experiment. The right panel is not presented as a direct recomputation of the left-panel point deviations. The available figure does not specify the noise channel, error reference and sampling repetitions, so it is an illustration rather than a quantitative noise-robustness benchmark. A controlled comparison would hold circuit parameters and decoder fixed and compare exact expectations, finite-shot estimates and noisy-device estimates.

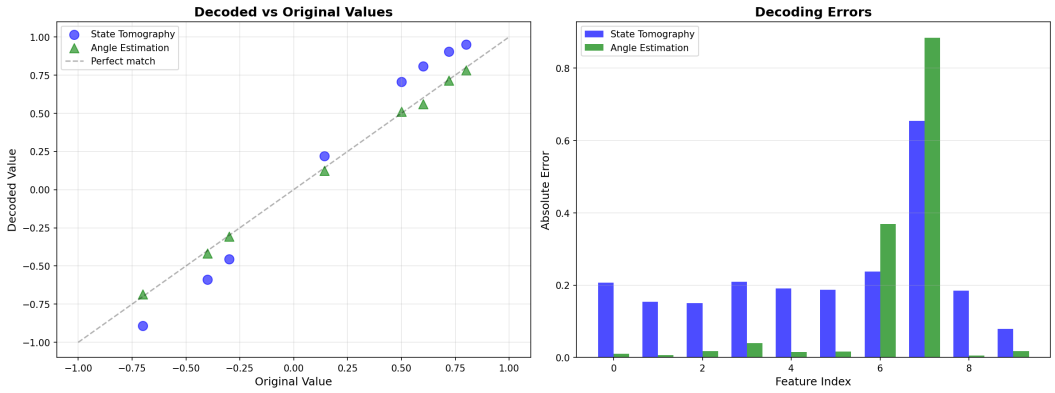


Figure 6: Decoding diagnostic. Left: original versus decoded values. Right: noise-related decoding errors in the same setup. The panels address different comparisons; their legends retain the experiment's terminology. The plot is supplementary and does not establish preservation of counterfactual quality under noise.

The hardware experiments were executed on `ibm_fez` using circuits ranging from 2 to 150 qubits, with 8192 shots per execution. For the two-qubit Moons example, the target-valid output $(1, 1)$ is 1.87345 L2 units from the factual input $(-0.2864, -0.3620)$. This example illustrates the end-to-end measurement-based generation procedure, while the larger circuits demonstrate that the proposed encoding and readout can be executed at substantially larger register sizes on real quantum hardware. These experiments establish hardware executability, but they do not constitute a matched simulator–device study or show that the multi-dataset initialization results are preserved under hardware noise.

E Supplementary Search Studies

These studies examine how initialization interacts with a subsequent search policy, and whether quality optimization preserves validity. Their configurations and single-run results are separate from the three-seed comparisons in Tables 1–2.

E.1 Protocol

Each study processes 50 factuals per dataset: Moons, Iris, Wine and Diabetes. All features are continuous. Moons contains 500 observations with noise 0.1 and an 80/20 stratified split. Iris uses a 60/40 split; Wine and Diabetes use 70/30. Iris and Wine map original class zero to binary label one and the other classes to zero. Diabetes thresholds its regression target at the dataset median. Standardization is fitted on training data, followed by division by the largest absolute standardized training value. A 100-tree random forest without a depth cap is fitted once, with split and forest random state 42. Up to 25 test examples per observed class are selected in order; additional examples from the other class fill any shortfall to 50.

The target is $t = 1 - f(x)$, including factuals whose prediction differs from their observed label. Saved class-aware directional parameters initialize each search; the source-to-target parameter set is selected anew for every instance. Circuit depth is two on Moons and five on the other datasets. Test-instance optimization does not update the saved starting sets. Search uses the continuous terms of Equations 5–6, with $\alpha_1 = 0.2$, $\eta_1 = \eta_2 = 0.8$ and $B_1 = B_2 = 0$. Stage limits are 150 and 75, using exact statevector evaluation.

The initializer comparison uses stochastic feature-space refinement; the stage study uses the coordinate refiner in Appendix A.4. Search random seeds are not recorded, so counts and means below are descriptive single-run results without uncertainty estimates. Initializer effort, offline parameter learning and circuit evaluations are not matched. The saved records do not identify checkpoint training examples, limiting verification of checkpoint provenance. These are supplementary mechanism and configured-pipeline studies rather than matched-budget generalization evidence.

E.2 Stage retention and optimizer sensitivity

The stage study records validity before optimization, after each circuit-optimization stage, and after coordinate refinement. It compares L-BFGS-B and COBYLA with the standard quality objective, and L-BFGS-B with a validity-guided retention variant. L-BFGS-B uses the parameter bounds in the method; the COBYLA runner does not impose those bounds. Shared stage limits therefore define optimizer configurations, not equal objective-evaluation budgets.

On Moons, Iris and Wine, retention adds $2C_t^{\text{invalid}}$, where $C_t^{\text{invalid}} = 2 + 4(0.5 - p_t)$ for $p_t < 0.5$ and zero otherwise. On Diabetes, it instead adds $30[0.5 - p_t]_+^2$. Both variants cache the nearest valid candidate encountered and return it when available. These different penalties are kept explicit rather than pooled as one fixed treatment. A further Diabetes control uses the same quadratic penalty without candidate retention.

Dataset	Configuration	Init	Step 1	Step 2	Final	Lost
Moons	L-BFGS-B / standard	34	34	12	48	22
Moons	COBYLA / standard	34	48	3	34	45
Moons	L-BFGS-B / retain valid	34	34	34	48	0
Iris	L-BFGS-B / standard	0	42	1	22	41
Iris	COBYLA / standard	0	49	13	31	36
Iris	L-BFGS-B / retain valid	0	42	45	50	0
Wine	L-BFGS-B / standard	8	19	12	29	7
Wine	COBYLA / standard	8	23	16	23	7
Wine	L-BFGS-B / retain valid	8	19	21	41	0
Diabetes	L-BFGS-B / standard	50	50	13	49	37
Diabetes	COBYLA / standard	50	50	22	50	28
Diabetes	L-BFGS-B / retain valid	50	50	50	50	0
Diabetes	L-BFGS-B / penalty only	50	50	26	50	24

Table 5: Complete stage study. Init, Step 1, Step 2 and Final are valid counts out of 50. Lost counts Step-1-valid instances that become invalid at Step 2. Retention uses the earlier penalty on Moons/Iris/Wine and the quadratic penalty on Diabetes. The penalty-only control was run on Diabetes.

Standard L-BFGS-B Step 2 reduces validity from 42/50 to 1/50 on Iris and from 50/50 to 13/50 on Diabetes. Retention avoids losing any Step-1-valid output in all four runs. Final validity increases from 22/50 to 50/50 on Iris and from 29/50 to 41/50 on Wine; it remains 48/50 on Moons and increases from 49/50 to 50/50 on Diabetes. The main-text panels show Iris and Wine; Table 5 includes every dataset and optimizer configuration.

The Diabetes control separates penalty guidance from candidate selection within the quadratic formulation: penalty alone leaves 26/50 outputs valid after Step 2, whereas penalty plus retention preserves 50/50. Both reach 50/50 after coordinate refinement. Retention changes intermediate validity in this run, without an observed final-success improvement over penalty alone. A retention-only control and matched-cost repeats remain necessary to isolate all component effects.

The cost measurements qualify the retention result. On Iris, retention adds approximately 1978 classifier calls in Step 2 while reducing mean refinement calls from 671 to 218. On Diabetes, penalty plus retention uses approximately 2370 Step-2 classifier calls versus 1185 for penalty alone. Retention also evaluates the circuit again to check and cache candidates; complete circuit counts are not recorded. These are reliability–cost trade-offs, not evidence of lower total computation. COBYLA attains higher Step-1 validity on Moons, Iris and Wine, but standard Step 2 can remove these gains. A successful validity stage is insufficient unless subsequent stages preserve or recover validity.

Dataset	Configuration	Step 1	Step 2	Refine	Valid L2
Moons	L-BFGS-B / standard	225.1	0.0	265.8	0.543
Moons	COBYLA / standard	49.8	0.0	325.6	0.391
Moons	L-BFGS-B / retain valid	225.1	392.8	150.5	0.730
Iris	L-BFGS-B / standard	388.3	0.0	671.2	0.348
Iris	COBYLA / standard	138.4	0.0	508.5	0.410
Iris	L-BFGS-B / retain valid	388.3	1978.2	218.1	0.526
Wine	L-BFGS-B / standard	1075.1	0.0	1559.5	0.364
Wine	COBYLA / standard	105.6	0.0	1757.8	0.385
Wine	L-BFGS-B / retain valid	1075.1	1688.7	1078.8	0.472
Diabetes	L-BFGS-B / standard	990.7	0.0	742.6	0.318
Diabetes	COBYLA / standard	91.9	0.0	618.1	0.341
Diabetes	L-BFGS-B / retain valid	990.7	2369.8	460.8	0.331
Diabetes	L-BFGS-B / penalty only	990.7	1184.9	469.8	0.340

Table 6: Mean classifier calls per stage and final L2 among *valid* outputs in the stage study. Zero Step-2 classifier calls means that its objective omits classifier queries, not zero circuit evaluations. Stage calls exclude diagnostic metric checks and offline training.

E.3 Component ablation

We further examine the contribution of learned directional initialization on Wine using 50 factual instances, 13 features, circuit depth $p = 5$, and a shared refinement budget of 150. We compare full QIOCE with the same pipeline using randomized initialization (QIOCE_RandInit), direct optimization (DirectOptim), and genetic search. Table E.3 reports validity before and after refinement together with the refined distance.

Method	Raw validity ↑	Refined validity ↑	Refined distance ↓
QIOCE	**38%**	**84%**	0.781
QIOCE_RandInit	22%	70%	**0.633**
DirectOptim	22%	76%	0.698
Genetic	22%	74%	0.667

Table 7: Component ablation on Wine ($n = 50$, 13 features, $p = 5$, refinement budget 150). Raw and Refined report target-class validity before and after shared refinement, respectively. Lower refined distance is better.

QIOCE attains the highest raw validity (38% versus 22% for all three alternatives) and the highest refined validity (84%). The refined-validity margin is smaller, with DirectOptim and Genetic reaching 76% and 74%, respectively, and QIOCE does not achieve the lowest refined distance. This ablation therefore provides evidence for improved target-class reachability rather than dominance across all counterfactual-quality criteria.

F Supplementary Experimental Views

The main text retains both Digits applications, the stage comparison, and the stage-retention diagnostic. The following figures provide complementary views without being counted as independent experiments.

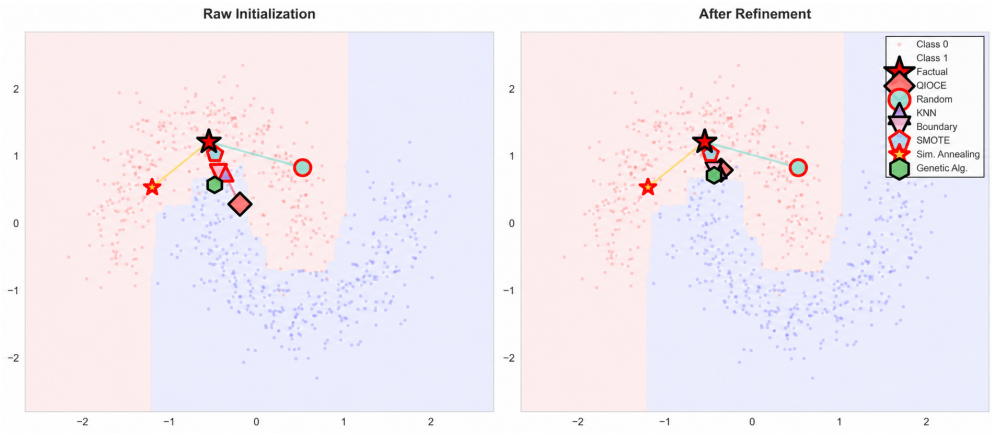


Figure 7: Candidate locations before and after refinement for one Moons factual. Different starting points lead local search to different parts of the decision boundary. This case illustrates the geometry behind the aggregate initialization comparison.

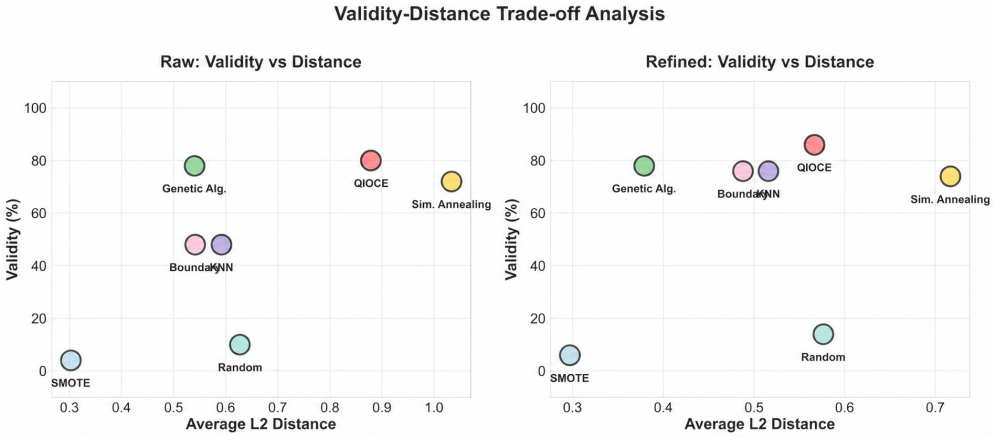


Figure 8: Validity–distance view of the Moons stage comparison in Figure 4. Higher validity and lower distance are preferred; the points show the trade-off between these objectives.

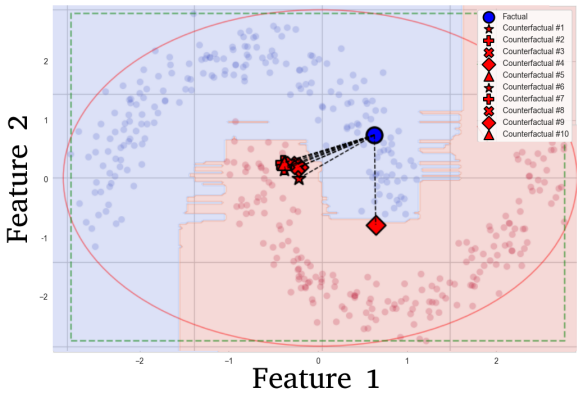


Figure 9: Multiple Moons alternatives for one factual input, illustrating non-uniqueness of prediction-changing outputs and their positions relative to data regions and the decision boundary.

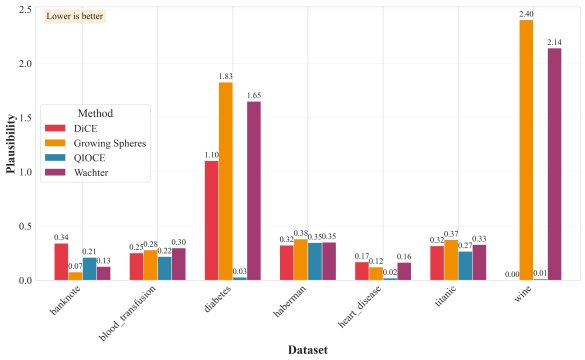


Figure 10: Visualization of the tabular plausibility scores. Interpretation follows the evaluator-assignment limitation in Appendix C.2

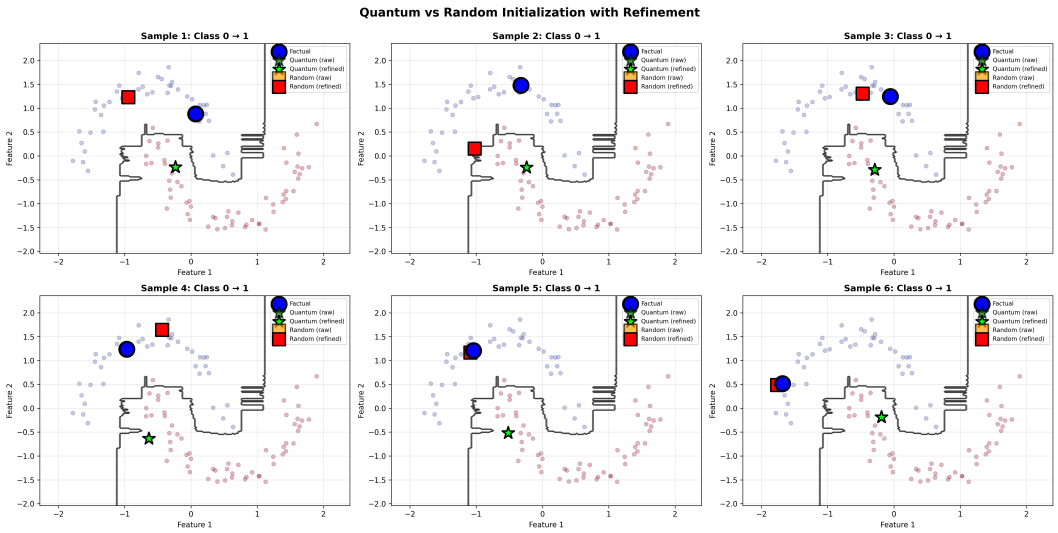


Figure 11: Additional quantum-versus-random initialization examples after refinement, showing candidate placement relative to a classifier boundary.