Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

QUBO-EASE: Per-Term Ensemble Weighting as a Quantum-Ready Binary Optimisation Problem for Hierarchical Protein Function Prediction

Farzad Majidi

Programming.Majidi@gmail.com

Islamic Azad University

Anvar Bahrampour

Islamic Azad University

Research Article

Keywords: protein function prediction, Gene Ontology, QUBO, ensemble learning, quantum annealing, label co-occurrence, matched controls

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11685954/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

QUBO-EASE: Per-Term Ensemble Weighting as a Quantum-Ready Binary Optimisation Problem for Hierarchical Protein Function Prediction

Farzad Majidi[1*], Anvar Bahrampour[1]

[1] Department of Software Engineering, Islamic Azad University, Sanandaj Branch, Sanandaj, Iran

* Corresponding author's Email: Programming.Majidi@gmail.com

Abstract

Quantum machine learning models for protein function prediction must compress protein language model embeddings to fit a few qubits, and matched-control studies have found no standalone advantage for them. We move the quantum-compatible component from feature processing to ensemble construction. In the proposed framework, QUBO-EASE, five classical base models are trained on full 1280-dimensional ESM-2 embeddings. For each Gene Ontology term, combination weights are obtained by minimising a quadratic energy built from weighted error correlation, pairwise synergy and model merit, encoded with three bits per model (15 binary variables per term). The weighted scores are refined by multi-channel fusion, a closed-form label co-occurrence model (EASE), logistic calibration, prior adjustment and per-protein normalisation. On 40,000 CAFA6 proteins and 180 terms, with a random 70/15/15 split and one seed, the framework reached an aspect-mean maximum F-measure of 0.5592, above logistic stacking (0.5479), simple averaging (0.5280), XGBoost (0.5252), a reimplemented DeepGOPlus (0.5236) and 14 other methods; the reported paired bootstrap intervals exclude zero. Removing calibration and post-processing lowered the score by 0.0147, and replacing exhaustive search with simulated annealing lowered it by 0.0040. A 10-qubit quantum neural network did not differ significantly from its parameter-matched control (+0.0043), and a ZZ quantum kernel scored below an RBF kernel (−0.0199). The QUBO instances were solved by classical exhaustive search, no quantum hardware or simulator was used for them, and the separate contributions of QUBO weighting and EASE were not isolated. We attribute the gain to the pipeline as a whole and not to quantum computation.

Keywords: protein function prediction; Gene Ontology; QUBO; ensemble learning; quantum annealing; label co-occurrence; matched controls

1. Introduction

Gene Ontology (GO) annotation assigns to each protein a set of terms describing its molecular functions, the biological processes it takes part in and the cellular components where it acts (Ashburner et al. 2000; Gene Ontology Consortium 2023). Experimental evidence covers only a small fraction of known sequences, so computational annotation is routinely used and is assessed in the Critical Assessment of Functional Annotation (CAFA) challenges (Radivojac et al. 2013; Jiang et al. 2016; Zhou et al. 2019). The task is a multi-label classification problem over a sparse, long-tailed and hierarchical label space, and a recurring observation of the CAFA assessments is that no single predictor is best across all terms and all three GO aspects.

Methods for this task fall into three broad families. Homology-based methods transfer the annotations of similar sequences found by BLAST or DIAMOND (Altschul et al. 1997; Buchfink et al. 2015). Classical machine learning methods combine engineered sequence features with nearest-neighbour voting or learning to rank (Liu et al. 2020; You et al. 2018). Deep learning methods learn from raw sequence or from protein language model embeddings (Kulmanov and Hoehndorf 2020; Cao and Shen 2021; Rives et al. 2021; Lin et al. 2023; Wang et al. 2023; Yuan et al. 2023). The strongest sequence-only systems are themselves combinations: DeepGOPlus mixes a convolutional network with a DIAMOND score, and TALE+ mixes a transformer with the same score, each with a single mixing weight per aspect (Kulmanov and Hoehndorf 2020; Cao and Shen 2021).

Quantum machine learning has been proposed as a further source of nonlinear feature maps (Havlíček et al. 2019; Schuld and Killoran 2019), but near-term devices offer few qubits (Preskill 2018), so a 1280-dimensional embedding must be reduced to a handful of rotation angles before it enters a circuit. Benchmark studies report that standard quantum models do not outperform simple classical models under controlled comparison (Bowles et al. 2024), and trainability and kernel concentration impose further limits (McClean et al. 2018; Thanasilp et al. 2024). Two earlier studies by the first author on the same CAFA6 subset reached the same conclusion for protein function prediction: representation width explained about 85% of the gain of a dimension-matched quantum feature map (Majidi 2026a), and five quantum models each scored below a matched classical control, while a small late-fusion gain was consistent with complementary prediction errors and remained untested against a classical ensemble (Majidi 2026b).

These negative results motivate the design studied here. Instead of placing a quantum circuit on the path of high-dimensional data, we place a quantum-compatible formulation at the stage where the problem is naturally discrete, low-dimensional and quadratic: choosing the weights with which complementary classical models are combined. Ensemble selection was cast as quadratic unconstrained binary optimisation (QUBO) in QBoost (Neven et al. 2008; Neven et al. 2012) and applied to particle physics and to transcription-factor binding (Mott et al. 2017; Li et al. 2018), but only for single-label problems. We formulate one small QUBO instance per GO term, so that each function receives its own combination of base models, and we follow the weighting step with a post-processing stage that transfers score between statistically co-occurring terms using the closed-form EASE model from recommender systems (Steck 2019).

We state the scope of the claim at the outset. Each instance has 15 binary variables and is solved exactly by classical exhaustive search; no quantum hardware or simulator is used to solve it, and at this size a quantum solver could at best return the same solution. The term "quantum-ready" refers to the form of the problem, which maps directly onto annealers and QAOA once the number of models or bits grows (Kadowaki and Nishimori 1998; Farhi et al. 2014; Lucas 2014). The main contributions of this work are:

- **Contribution 1:** We introduce QUBO-EASE, a four-step pipeline that feeds the full 1280-dimensional ESM-2 embedding to five base models with different inductive biases and applies no dimensionality reduction.
- **Contribution 2:** We formulate per-term ensemble weighting as a binary quadratic problem whose energy combines class-balanced error correlation, pairwise synergy, a composite merit score and sparsity penalties, with a non-uniform three-bit weight encoding (15 binary variables per term).
- **Contribution 3:** We define a post-processing stage that combines four fusion channels, two EASE label co-occurrence channels, per-term logistic calibration, a prior exponent and per-protein normalisation, with every coefficient selected on a held-out validation half.
- **Contribution 4:** We compare the framework on one split and one scoring implementation with 18 other methods: seven configurations of published methods reimplemented on the same data, six classical models on ESM-2, two classical ensembles, two quantum models with matched classical controls, and the naive frequency baseline.
- **Contribution 5:** We report an ablation of the solver and of the post-processing stage with paired bootstrap intervals, and we state which hypotheses the experiments did not test.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed methodology. Section 4 describes the experimental setup and reports the results, the ablation and the complexity analysis. Section 5 discusses the findings and their limitations and concludes.

2. Related Work

Homology and classical machine learning. Transferring annotations from sequence neighbours remains a strong baseline. DiamondScore scores a term by the bit-score-weighted fraction of DIAMOND hits that carry it (Buchfink et al. 2015; Kulmanov and Hoehndorf 2020), and fails when a query has no sufficiently similar annotated sequence. GODoc uses k-nearest-neighbour voting on PSSM-derived frequencies reduced by principal component analysis and propagates scores along parent–child edges (Liu et al. 2020). GOLabeler and NetGO combine several component predictors with learning to rank (You et al. 2018; You et al. 2019); the second-level ranker does not model the error correlation of its components explicitly.

Deep learning and protein language models. DeepGO introduced an ontology-aware classifier on sequence and interaction data (Kulmanov et al. 2018). ProLanGO treats annotation as neural machine translation from k-mers to term codes (Cao et al. 2017). DeepGOPlus combines a flat convolutional classifier with DiamondScore using one weight per aspect (Kulmanov and Hoehndorf 2020), and TALE embeds sequences and labels jointly with a hierarchy regulariser, TALE+ adding the same homology score (Cao and Shen 2021). Protein language models such as ESM-1b and ESM-2 provide embeddings on which simple classifiers are competitive (Rives et al. 2021; Lin et al. 2023); NetGO 3.0 adds an ESM-1b component to a ranking ensemble (Wang et al. 2023), SPROF-GO diffuses labels over a protein similarity graph (Yuan et al. 2023), and DeepGO-SE uses ontology axioms (Kulmanov et al. 2024). Appel and Butcher (2026) report gradient boosting on pretrained embeddings, which we include as a baseline.

Label dependence. Existing methods either predict terms independently, use only the explicit parent–child structure of the ontology, or propagate over proteins and not over terms. The statistical co-occurrence of terms is not used directly. EASE is a linear autoencoder with a closed-form solution and a zero-diagonal constraint, proposed for collaborative filtering (Steck 2019); its weight matrix can be read as a co-occurrence graph between items. To our knowledge it has not been applied to transfer scores between GO terms.

Ensembles and QUBO. Combining diverse models helps when their errors are not strongly correlated (Krogh and Vedelsby 1995), and stacking learns the combination from held-out predictions (Wolpert 1992; Caruana et al. 2004). QBoost selects a subset of weak classifiers by minimising a quadratic objective in which linear terms reward agreement with the label and quadratic terms penalise correlated members (Neven et al. 2008; Neven et al. 2012). Mott et al. (2017) solved such a problem on a quantum annealer for Higgs event classification and Li et al. (2018) for transcription-

factor binding; both found a relative benefit only for small training sets. These studies are single-label, and their conclusions rest on comparison with classical solvers such as simulated annealing (Kirkpatrick et al. 1983).

Quantum machine learning for this task. Bowles et al. (2024) showed that standard quantum classifiers do not outperform simple classical baselines under matched benchmarking. On the CAFA6 subset used here, a dimension-matched quantum feature map and five further quantum models gave no standalone advantage (Majidi 2026a; Majidi 2026b). Table 1 summarises the representative studies.

Table 1. Comparison of representative studies. Reported outcomes are those stated by the cited authors.

Reference	Year	Method	Dataset	Advantage	Limitation
(Neven et al. 2012)	2012	QBoost: binary selection of weak classifiers as QUBO	Synthetic and image benchmarks	Quadratic terms penalise correlated members; compact ensembles	Single-label; binary inclusion weights
(Mott et al. 2017)	2017	QUBO ensemble of 36 weak classifiers on a quantum annealer	Higgs signal vs background	Robust for small training sets	Behind boosted trees with more data
(Li et al. 2018)	2018	Annealing-based classifier	Transcription-factor binding	Relative benefit with little data	Single-label; classical methods better at scale
(You et al. 2018)	2018	GOLabeler: learning to rank over component predictors	CAFA-style benchmark	Combines several sequence sources	No explicit model of error correlation
(Kulmanov and Hoehndorf 2020)	2020	DeepGOPlus: CNN plus DiamondScore	CAFA3	F_max 0.544 (MF), 0.469 (BP), 0.623 (CC)	Two members; one weight per aspect; flat classifier
(Liu et al. 2020)	2020	GODoc: kNN voting on TFPSSM with PCA	CAFA3	F_max 0.592 in CC	Dimensionality reduction; one model family
(Cao and Shen 2021)	2021	TALE+: transformer with joint sequence–label embedding plus DiamondScore	CAFA3	F_max 0.615 (MF), 0.431 (BP), 0.669 (CC)	Explicit hierarchy only; one weight per aspect
(Wang et al. 2023)	2023	NetGO 3.0: ESM-1b component in a ranking ensemble	CAFA-style benchmark	Language model embeddings without reduction	Second-level ranker; no per-term weighting
(Yuan et al. 2023)	2023	SPROF-GO: ProtT5 with homology-based label diffusion	Own benchmark	Fast sequence-only prediction	Diffusion over proteins, not over terms
(Bowles et al. 2024)	2024	Benchmark of standard quantum classifiers	Binary classification benchmarks	Controlled comparison with classical baselines	Benchmark study, not a predictor
(Majidi 2026a)	2026	Dimension-matched quantum feature map (QEPP)	CAFA6 subset, 180 terms	Attributes the gain to its sources	About 85% of the gain due to width; 49× time
(Majidi 2026b)	2026	Five quantum models with matched controls	CAFA6 subset, 180 terms	Late fusion 0.6697 vs RBF-kNN 0.6719	No standalone advantage; fusion untested against a classical ensemble
This work	2026	Per-term QUBO weighting of five classical models plus EASE post-processing	CAFA6 subset, 180 terms	Full embedding; per-term weights; label co-occurrence	Classical solver; one seed; random split

Three gaps follow from this review. First, quantum models must compress the input representation and show no matched advantage, so the embedding is better left to classical models. Second, the successful function predictors combine two sources with one weight per aspect, or use a second-level ranker, and none weights models per term while accounting for error correlation; QUBO ensemble methods do this but only in single-label settings. Third, statistical co-occurrence between terms is not exploited. QUBO-EASE addresses the three gaps together.

3. Proposed Methodology

3.1 Theoretical Background and Problem Formulation

Gene Ontology and the label space. GO terms form a directed acyclic graph in three aspects: molecular function (MF), biological process (BP) and cellular component (CC). Annotations are sparse and long-tailed: a few terms annotate many proteins and most terms annotate few. The protein-centric maximum F-measure used by CAFA applies one global threshold to all proteins, which makes it sensitive to the scale of the scores across proteins and across terms (Radivojac et al. 2013; Clark and Radivojac 2013).

QUBO. A QUBO problem minimises $b^\top Q_b\, b$ over binary vectors $b \in \{0,1\}^n$ and is equivalent to an Ising model, the native problem of quantum annealers (Lucas 2014; Glover et al. 2019). A real weight can be represented by a few binary variables with fixed coefficients. For ensemble construction, linear terms express the quality of individual members and quadratic terms their pairwise interaction, as in QBoost (Neven et al. 2012).

EASE. Given a binary matrix X, EASE minimises $\| X - XB \|_F^2 + \lambda \| B \|_F^2$ subject to $\mathrm{diag}(B) = 0$. The constraint excludes the trivial solution $B = I$, so each column must be predicted from the others, and the solution is available in closed form from the inverse of the regularised Gram matrix (Steck 2019). With proteins as users and GO terms as items, X is the training label matrix and B encodes which terms predict which others after the effect of the remaining terms is removed.

3.2 Problem definition

Let $x_p \in \mathbb{R}^{1280}$ be the ESM-2 embedding of protein p, T_p its set of recorded GO terms and $t_1, \ldots, t_C$ the selected terms, with $C = 180$. Labels are used as recorded in the annotation file, without re-propagation over the ontology:

$$y_{p,j} = \mathbb{I}\big(t_j \in T_p\big), \qquad j = 1, \ldots, C. \tag{1}$$

The task is to learn a scoring function

$$f: \mathbb{R}^{1280} \to [0,1]^C, \qquad \hat{y}_p = f\big(x_p\big), \tag{2}$$

that maximises the aspect-mean protein-centric $F_{\max}$ on unseen proteins. We write $M = 5$ for the number of base models, $s_{p,j}^{(m)} \in [0,1]$ for the score of model m, $a(j)$ for the aspect of term j, $w^{(j)} \in [0,1]^M$ for the combination weights of term j, $K = 3$ for the number of bits per weight and π_j for the prior of term j in the training data.

3.3 Overall architecture

Figure 1 shows the four steps and the data on which each component is fitted. Step 1 represents each protein by its frozen ESM-2 embedding. Step 2 trains five base models on these embeddings. Step 3 builds and solves one QUBO instance per term from statistics of the base models on the first half of the validation data. Step 4 builds fusion channels from the weighted scores, refines them with EASE and logistic calibration, and applies a prior exponent and per-protein normalisation.

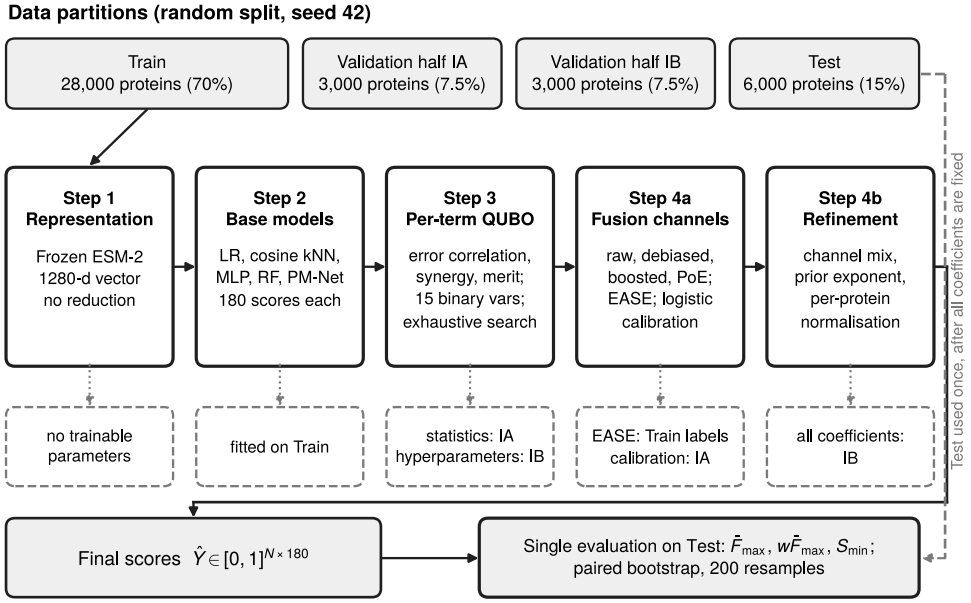


Fig 1. Data flow of the QUBO-EASE framework. Upper row: data partitions. Middle row: the four steps. Dashed boxes: the partition on which each component is fitted. The test split is used once, after every coefficient is fixed.

No component is fitted on the test split. The base models, the EASE matrices and the priors are fitted on the training split. The validation split is divided into two disjoint halves: IA is used to build the QUBO instances and to fit the calibrators, and IB is used to select hyperparameters. The QUBO statistics are computed on IA and not on the training split because the training scores of the nearest-neighbour and random-forest models are optimistic through memorisation and would bias the weights towards them.

3.4 Mathematical model

Base models. Four of the five base models are trained with a weighted binary cross-entropy in which the positive term of label j is multiplied by

$$\omega_j = \min\left(\max\left(\frac{N_j^-}{\max\left(N_j^+,1\right)},1\right),20\right), \tag{3}$$

where N_j^+ and N_j^- are the numbers of positive and negative training proteins; the cap of 20 limits the weight of rare terms. This weighting shifts the scores upwards, which Step 3 corrects. The models are: (i) multi-label logistic regression (LR), one linear layer from 1280 inputs to 180 sigmoid outputs; (ii) cosine nearest neighbours (Cos-kNN) with $k = 15$, whose score is the similarity-weighted mean of the neighbours' labels,

$$s_{p,j}^{(2)} = \frac{\sum_{q\in\mathcal{N}_{15}(p)} \max(\mathrm{sim}(x_p,x_q),0)\,y_{q,j}}{\sum_{q\in\mathcal{N}_{15}(p)} \max(\mathrm{sim}(x_p,x_q),0)+\epsilon}; \tag{4}$$

(iii) a multilayer perceptron (MLP) with one hidden layer of 512 units, layer normalisation, GELU and dropout 0.15; (iv) a multi-output random forest (RF) with 30 trees, maximum depth 10 and at least five samples per leaf (Breiman 2001), fitted on a random subsample of 12,000 training proteins; and (v) a parameter-matched network (PM-Net). PM-Net is the classical control of the quantum neural network used in Section 4: it has the same architecture with the circuit replaced by a classical layer of equal size (Majidi 2026b). A two-layer trunk maps the input to $h_p \in \mathbb{R}^{256}$, after which

$$\theta_p = \pi\tanh\left(W_\theta h_p + c_\theta\right) \in [-\pi,\pi]^{20}, \qquad q_p = \tanh\left(W_q\theta_p + c_q\right) \in \mathbb{R}^{165}, \tag{5}$$

and $[h_p; q_p]$ is passed to a 512-unit head with 180 outputs. In the quantum model, θ_p holds two rotation angles for each of 10 qubits and q_p is replaced by 165 Pauli expectation values (three measurement bases times 10 single-qubit and 45 two-qubit observables). PM-Net has 1,106,587 parameters against 1,103,182 for the quantum network. Table 2 lists the base models.

Table 2. Base models of the framework, as executed.

Model	Architecture	Trainable parameters	Training
LR	Linear 1280 → 180, sigmoid	230,580	30 epochs; learning rate 0.003; weighted BCE
Cos-kNN	k = 15, cosine similarity, negative similarities clipped	none	Stores training embeddings and labels
MLP	1280 → 512 → 180; layer norm, GELU, dropout 0.15	749,236	25 epochs; learning rate 0.002; weighted BCE
RF	30 trees, depth ≤ 10, ≥ 5 samples per leaf	n/a	12,000-protein subsample of the training split
PM-Net	Trunk 1280 → 512 → 256; 20 angles → 165 features; head 512 → 180	1,106,587	20 epochs, patience 5; learning rate 0.002; dropout 0.15; EMA 0.995

Score preparation. A model whose mean score on IA exceeds 2.5 times the mean prior is treated as positively weighted and is debiased in logit space,

$$\tilde{s}_{p,j}^{(m)} = \sigma\left(\mathrm{logit}\left(s_{p,j}^{(m)}\right) - \log\omega_j\right), \tag{6}$$

which cancels the effect of ω_j on the logit; for the other models $\tilde{s} = s$. The logits of the debiased scores of each model are standardised on IA and denoted $z_{p,m}$.

Weighted error correlation. For term j, the standardised error of model m is $\varepsilon_{p,m} = z_{p,m} - \left(y_p - \bar{y}\right)$ with $\bar{y} = \sum_p \Omega_p\, y_p$. Samples are weighted so that positives and negatives each carry half of the total weight,

$$\Omega_p = \begin{cases} \frac{1}{2N_j^+} & y_{p,j} = 1 \\ \frac{1}{2N_j^-} & y_{p,j} = 0 \end{cases} \tag{7}$$

and the error correlation matrix follows from the weighted second moments:

$$\Sigma_{mm'} = \sum_{p\in\mathrm{IA}} \Omega_p\ \varepsilon_{p,m}\varepsilon_{p,m'}, \qquad R_{mm'} = \frac{\Sigma_{mm'}}{\sqrt{\Sigma_{mm}\Sigma_{m'm'}}}. \tag{8}$$

A large positive $R_{mm'}$ means that two models err on the same proteins and that combining them gains little.

Merit and target weights. Four indicators are computed on IA for each model: its single-model $F_{\max}$ in the aspect, F_m^a; its leave-one-out contribution Δ_m^a, the difference in $F_{\max}$ between the mean of all models and the mean without model m; its AUPR on term j; and its maximum F1 on term j over 18 thresholds. Each indicator is rescaled to [0,1] across the five models (denoted by a hat), and the composite merit is

$$\mu_m = 0.35\,\hat{F}_m^a + 0.30\,\hat{\Delta}_m^a + 0.20\,\widehat{\mathrm{AUPR}}_{mj} + 0.15\,\widehat{\mathrm{F1}}_{mj}, \qquad \bar{\mu}_m = \frac{\mu_m^\gamma}{\sum_{m'}\mu_{m'}^\gamma}. \tag{9}$$

The target weight vector mixes the merit with the normalised positive coefficients w^{LR} of a stacking logistic regression fitted on IA:

$$t \propto \big((1-\alpha)\bar{\mu} + \alpha w^{\mathrm{LR}}\big) \odot (\mathbf{1} - 0.85\,h), \qquad h_m = \mathbb{I}(\Delta_m^a < -0.0005), \qquad \ell_m = 1 - \frac{t_m}{\max_{m'} t_{m'}}. \tag{10}$$

Here h flags harmful models, whose removal raises the aspect $F_{\max}$, and ℓ_m is the distance of model m from the best model in target weight.

Pairwise synergy. The synergy of two models in aspect a is the gain of their best weighted pair over the better of the two, measured on IA:

$$G_{mm'} = \max_{v \in \{0.3,0.5,0.7\}} F_{\max}^a\big(v\tilde{S}^{(m)} + (1-v)\tilde{S}^{(m')}\big) - \max(F_m^a, F_{m'}^a). \tag{11}$$

G is normalised by its largest absolute entry to give $\hat{G}$. Unlike R, which is computed per term, G is computed per aspect and shared by the terms of that aspect, because $F_{\max}$ is protein-centric and is not defined for a single term.

Energy function. The matrix and vector of the energy of term j are

$$Q_{mm'} = \frac{0.45\,R_{mm'} - \beta\,\hat{G}_{mm'} + 0.65\,\ell_m \ell_{m'}}{M^2} \;(m \neq m'), \qquad Q_{mm} = \frac{1.6 + 3.8\,\ell_m^{1.6} + 2.5\,h_m}{M^2}, \qquad q_m = -\frac{3.2}{M}\,t_m, \tag{12}$$

and the energy minimised for each term is

$$E(w) = w^{\top} Q\, w + q^{\top} w + \lambda_1 \mathbf{1}^{\top} w + \lambda_0 \parallel w \parallel_0, \qquad w \neq 0. \tag{13}$$

The linear term rewards models with a high target weight. In the quadratic term, $R_{mm'}$ penalises pairs with correlated errors, $-\beta \hat{G}_{mm'}$ rewards complementary pairs and $\ell_m \ell_{m'}$ penalises the joint presence of two weak models. The diagonal entries are the cost of including a model and are larger for weak and harmful models. The last two terms make the weights sparse. The fixed coefficients in Eqs. (9)–(12) were set empirically before the final run, and the five hyperparameters α, β, γ, λ_1 and λ_0 are selected per aspect on IB.

Three-bit encoding. Each weight is encoded by three bits with unequal values,

$$w_m = \textstyle\sum_{k=0}^{2} v_k\; b_{m,k}, \qquad v = (0.06,\, 0.20,\, 0.74), \qquad b_{m,k} \in \{0,1\}, \tag{14}$$

so that it takes one of the eight values $\{0, 0.06, 0.20, 0.26, 0.74, 0.80, 0.94, 1\}$. Compared with a uniform binary encoding these values are denser near zero, which lets a model enter the combination with a very small weight. Five models require $MK = 15$ binary variables per term (Figure 2). Writing $w = Vb$ with a block matrix $V \in \mathbb{R}^{5\times 15}$ and using $b_i^2 = b_i$, the first three terms of Eq. (13) give the standard QUBO form

$$Q_b = V^{\top} Q\, V + \mathrm{diag}\big(V^{\top}(q + \lambda_1 \mathbf{1})\big), \qquad \min_{b \in \{0,1\}^{15}} b^{\top} Q_b\, b. \tag{15}$$

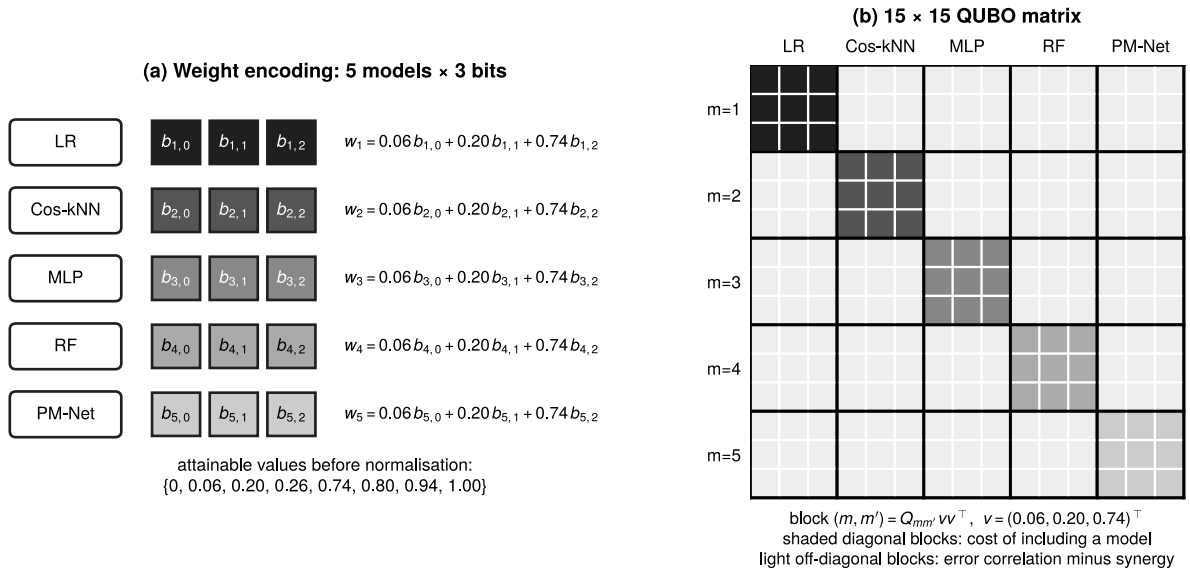


Fig 2. (a) Three-bit encoding of the weights of the five base models in 15 binary variables for one term. (b) Block structure of the resulting QUBO matrix.

The term $\lambda_0 \parallel w \parallel_0$ is cubic in the bits, because model m is inactive only when its three bits are all zero. With $\lambda_0 > 0$ the problem is therefore a higher-order binary optimisation problem and not strictly a QUBO. The exhaustive solver evaluates this term directly; the simulated-annealing solver, which accepts only the quadratic form of Eq. (15), omits it. After solving, the weights are normalised to unit sum, $\hat{w} = w^* / \mathbf{1}^{\top} w^*$.

Solvers. With 15 binary variables each term has $2^{15} = 32{,}768$ states. The main solver enumerates all states, vectorised over the terms of an aspect, and returns the global minimum; the all-zero vector is excluded. The second solver is simulated annealing (Kirkpatrick et al. 1983) with 220 temperature steps from 0.45 to 10^{-4} on a geometric

schedule and six random restarts, used only in the ablation. At this problem size a quantum solver has no advantage over exhaustive search and can at best return the same solution, so any improvement shown by the framework is attributed to the formulation and the pipeline and not to a quantum solver.

Fusion channels. With the normalised weights, four channels are built for each protein and term:

$$r = \sum_m \hat{w}_m\, s^{(m)}, \qquad d = \sum_m \hat{w}_m\, \tilde{s}^{(m)}, \qquad b = \sum_m \hat{w}_m\, \check{s}^{(m)}, \qquad u = \sigma\big(\textstyle\sum_m \hat{w}_m\ \mathrm{logit}\, \tilde{s}^{(m)}\big). \tag{16}$$

Here r is the raw channel, d the debiased channel and u a product-of-experts channel in logit space (Hinton 2002). The boosted channel b keeps $\check{s} = s$ for positively weighted models and shifts the other models upwards to a comparable scale:

$$\check{s} = 0.55\, \sigma\big(\mathrm{logit}\big(s + 0.005\, \pi_j\big) + 0.85 \log \omega_j\big) + 0.45\, s^{0.55}. \tag{17}$$

EASE channels. With Y the training label matrix, the EASE solution is

$$P = (Y^\top Y + \lambda I)^{-1}, \qquad \hat{B}_{ij} = \begin{cases} 0 & i = j \\ -P_{ij}/P_{jj} & i \neq j \end{cases} \tag{18}$$

We fit a sharp matrix with $\lambda = 80$ and a broad matrix with $\lambda = 350$ and set negative entries to zero, so that score is transferred only from positively co-occurring terms. The two EASE channels are

$$e_p^{(1)} = \mathrm{clip}\big(d_p \hat{B}_{80}, 0, 1\big), \qquad e_p^{(2)} = \mathrm{clip}\big(b_p \hat{B}_{350}, 0, 1\big), \tag{19}$$

where d_p and b_p are the row vectors of the debiased and boosted channels over the 180 terms.

Logistic calibration. For each term a small logistic regression on the four features $\big(d, \sqrt{d}, b, e^{(1)}\big)$ is fitted on IA and yields a calibrated probability c. This generalises Platt scaling (Platt 1999) and brings the sharp EASE channel into the score in a learned way. The inverse regularisation strength is $\min\big(\max\big(N_j^+/35, 0.15\big), 1.2\big)$, so rare terms are regularised more strongly; for terms with fewer than two positives in IA, $c = d$.

Channel mixing, prior and per-protein normalisation. The mixed score is

$$g = \frac{\theta\, c + (1-\theta)\, \rho + \eta\big(0.6\, e^{(1)} + 0.4\, e^{(2)}\big)}{1+\eta}, \tag{20}$$

where the base channel ρ is one of d, b, r, u, $\frac{1}{2}(d+r)$ and $\frac{1}{2}(b+r)$, $\theta \in \{0, 0.2, 0.4, 0.65, 0.85\}$ and $\eta \in \{0, 0.08, 0.18, 0.28\}$. With $\eta = 0$ the direct EASE term is removed, although EASE still acts through the calibrator. The prior, estimated with Laplace smoothing as $\pi_j = \big(N_j^+ + 1\big)/(N+2)$, is applied as $g'_{p,j} = g_{p,j}\, \pi_j^{\gamma_p}$ with $\gamma_p \in \{-0.06, 0, 0.05\}$, followed by division by the maximum in the aspect. Finally, because $F_{\max}$ uses a global threshold, a protein whose scores are low overall can have its correct terms fall below it. With m_p the maximum score of protein p in the aspect, or the mean of its two highest scores,

$$\hat{g}_{p,j} = m_p^{\alpha_p} \left(\frac{g'_{p,j}}{m_p}\right)^{\eta_p}, \qquad \hat{y}_{p,j} = \Big((1-\chi)\, g'_{p,j} + \chi\, \hat{g}_{p,j}\Big)^{\kappa}. \tag{21}$$

An exponent $\alpha_p < 1$ raises the maximum of low-scoring proteins towards one, η_p reshapes the scores of a protein relative to its maximum and χ sets the strength of the transform. After $\alpha_p \in \{0.25, \ldots, 1\}$, $\eta_p \in \{0.85, 1, 1.15, 1.35\}$ and $\chi \in \{0.25, 0.5, 0.75, 1\}$ are selected, the power $\kappa = \log 0.38 / \log \tau$, clipped to [0.55, 1.85], moves the optimal IB threshold τ to about 0.38; if this lowers $F_{\max}$ on IB, $\kappa = 1$. All coefficients of this stage are selected per aspect by maximising $F_{\max}$ on IB.

3.5 Algorithm and workflow

Algorithm 1 gives the training and evaluation procedure and Algorithm 2 the construction and solution of the QUBO instances of one aspect. The test split enters only in line 15 of Algorithm 1, after all coefficients are fixed. In Algorithm 2 the outer loop runs over 27 combinations of (α, β, γ) and the inner loop over 16 combinations of (λ_1, λ_0), so 432 configurations are evaluated per aspect and one is kept.

Algorithm 1. Training and evaluation of QUBO-EASE

Input: ESM-2 vectors $X \in \mathbb{R}^{N \times 1280}$, labels $Y \in \{0,1\}^{N \times 180}$, seed 42
Output: final scores $\hat{Y}$ on test proteins, configuration per aspect
1: split proteins 70/15/15 into train, validation, test; split validation into halves IA, IB
2: train LR, Cos-kNN ($k = 15$), MLP, RF (30 trees), PM-Net on train
3: $S^{(m)} \leftarrow$ scores of model m on validation and test, $m = 1, \ldots, 5$
4: $\tilde{S}^{(m)} \leftarrow$ debias positively weighted models (Eq. 6)
5: $\hat{B}_{80}, \hat{B}_{350} \leftarrow \mathrm{EASE}(Y_{\mathrm{train}})$; zero diagonal, clip negatives (Eq. 18)
6: **for** each aspect $a \in \{\mathrm{MF}, \mathrm{BP}, \mathrm{CC}\}$ **do**
7: $\quad \widehat{W}_a \leftarrow$ SolveAspectQUBO(a) (Algorithm 2)
8: **end for**

9: build channels $r, d, b, u, e^{(1)}, e^{(2)}$ with weights $\widehat{W}$ (Eqs. 16–19)
10: **for** $j = 1, \ldots, 180$ **do** fit the four-feature logistic calibrator on IA $\rightarrow c_j$
11: **for** each aspect a **do**
12: select θ, base channel ρ, η, γ_p by $F_{\max}$ on IB (Eq. 20)
13: select α_p, η_p, χ and the top-k rule on IB; set κ (Eq. 21)
14: **end for**
15: apply the selected configuration to the test scores $\rightarrow \hat{Y}$
16: **return** $\hat{Y}$; evaluate $\bar{F}_{\max}$ with the paired bootstrap (200 resamples)

Algorithm 2. Construction and solution of the QUBO instances of aspect a

Input: debiased validation scores $\tilde{S}$ (IA, IB), labels, terms J_a of aspect a
Output: normalised weight matrix $\widehat{W}_a \in [0,1]^{60\times 5}$
1: on IA: F_m^a, leave-one-out gain Δ_m^a, pairwise synergy G (Eq. 11)
2: **for** each term $j \in J_a$ **do** R_j, AUPR, F1, w_j^{LR} on IA (Eq. 8)
3: best $\leftarrow -\infty$
4: **for** (α, β, γ) in the 3 × 3 × 3 grid **do**
5: **for** each term j **do** build Q_j, q_j from $\mu, t, \ell, h, R_j, \hat{G}$ (Eqs. 9–12)
6: **for** (λ_1, λ_0) in the 4 × 4 grid **do**
7: **for** each term j **do** $w_j \leftarrow \operatorname{argmin} E(w)$ over 32,768 states, $w \neq 0$ (Eq. 13)
8: $f \leftarrow F_{\max}$ of the simple aspect score on IB
9: **if** $f >$ best **then** best $\leftarrow f$; store $(\alpha, \beta, \gamma, \lambda_1, \lambda_0)$ and W
10: **end for**
11: **end for**
12: **return** $\widehat{W}_a \leftarrow$ rows of the stored W normalised to unit sum

The grids are $\alpha \in \{0.25, 0.5, 0.75\}$, $\beta \in \{0.6, 1.5, 2.6\}$, $\gamma \in \{2.5, 4, 5.5\}$, $\lambda_1 \in \{0, 0.01, 0.03, 0.07\}$ and $\lambda_0 \in \{0, 0.015, 0.04, 0.08\}$. The simple aspect score used in line 8 of Algorithm 2 is $0.82\, d + 0.18\, e^{(1)}$ for MF, the mean of the boosted and raw channels for BP and the debiased channel for CC.

4. Experimental Setup

4.1 Dataset description

The data come from the training release of the CAFA 6 challenge (Iowa State University et al. 2025): annotations from train_terms.tsv and sequences from train_sequences.fasta. The label space contains 180 terms, the 60 terms with the most annotated proteins in each aspect, which matches the size used in earlier studies on these data (Majidi 2026a; Majidi 2026b). From the proteins that carry at least one of these terms and have an ESM-2 embedding, 40,000 are sampled at random with seed 42 and split by a random permutation into training, validation and test parts (Table 3).

Table 3. Dataset summary.

Item	Value
Source	CAFA 6 training release (train_terms.tsv, train_sequences.fasta)
Proteins	40,000, sampled at random (seed 42)
Train / validation / test	28,000 / 6,000 / 6,000 (70/15/15, random)
Validation halves	IA: 3,000 (QUBO statistics, calibration); IB: 3,000 (hyperparameter selection)
Label space	180 GO terms: 60 most frequent per aspect (MF, BP, CC)
Labels	As recorded in the annotation file; no re-propagation over the ontology
Representation	ESM2-t33-650M, mean-pooled, 1280 dimensions, frozen, loaded from a public precomputed set

The split is random and not cluster-disjoint, so homologous proteins can appear in both the training and the test parts. This favours models that transfer the labels of similar proteins, such as nearest neighbours. Because every method is evaluated on the same split the relative comparison is not made unfair by it, but the absolute $F_{\max}$ values may be higher than they would be on entirely novel proteins. Restricting the label space to frequent terms concentrates the

evaluation on terms with enough samples for stable estimates of the model statistics; the results do not extend to rare terms.

4.2 Implementation details

Table 4 records the configuration as executed. The framework is implemented in Python in a Kaggle notebook with a GPU. The neural models, the nearest-neighbour model and the quantum neural network use PyTorch (Paszke et al. 2019); the random forest, the calibration and stacking regressions and AUPR use scikit-learn (Pedregosa et al. 2011); the QUBO construction, the exhaustive search and simulated annealing are vectorised in NumPy. Table 5 gives the post-processing configuration selected on IB for each aspect.

Table 4. Implementation configuration, as executed.

Component	Setting
Optimiser (neural models)	AdamW (Loshchilov and Hutter 2019); weight decay 10^{-5}; batch size 256; cosine schedule with 5% warm-up; gradient clipping at norm 5; best validation epoch kept
QUBO encoding	3 bits per model, 15 binary variables per term; exhaustive search over 32,768 states
QUBO hyperparameter grid	432 configurations per aspect (27 × 16), selected on IB
Simulated annealing (ablation)	220 steps, temperature 0.45 → 10^{-4} (geometric), 6 restarts
EASE	$\lambda = 80$ (sharp) and $\lambda = 350$ (broad); zero diagonal; negative entries set to zero
Quantum neural network	10 qubits, 3 layers, CZ gates in a linear topology, 165 Pauli expectations; 20 epochs; exact state-vector simulation
DeepGOCNN	Input length 2000; filter lengths 8–128, 512 filters each
TALE	Input length 1000; dimension 128; 2 layers; 3 seeds averaged for TALE+
DIAMOND	Official release; E-value threshold 0.001
Statistical test	Paired bootstrap over test proteins, 200 resamples, 95% percentile interval
Number of runs	One seed (42) for every method
Compute environment	Kaggle notebook with GPU

Table 5. Post-processing configuration selected on IB for each aspect.

Hyperparameter	MF	BP	CC
Calibration share θ	0.40	0.20	0.40
Base channel ρ	d	(b + r)/2	d
Direct EASE share η	0	0.28	0
Prior exponent γ_p	−0.06	−0.06	0
Normalisation (α_p, η_p, χ)	(0.25, 1.15, 1)	(0.25, 1.35, 1)	(0.25, 0.85, 1)
Power κ	0.62	1.52	1

Per-protein normalisation was selected at full strength with $\alpha_p = 0.25$ in all three aspects. The direct EASE share is non-zero only in BP, while the sharp EASE channel acts in all three aspects through the calibrator.

4.3 Baseline methods

Table 6 lists the compared methods in five groups. The literature methods were reimplemented on the same data. DiamondScore follows the bit-score-weighted definition with DIAMOND. DeepGOCNN uses the published architecture and DeepGOPlus combines it with DiamondScore; the mixing weights for DeepGOPlus and TALE+ were selected per aspect on the validation split, as in the original papers. TALE uses a transformer encoder with joint sequence–label embedding and the hierarchy penalty. ProLanGO is a three-layer LSTM encoder–decoder with attention, and the score of each term is taken from the decoder probabilities. GODoc could not be run with PSI-BLAST profiles in the execution environment, so spaced-dipeptide frequencies were computed from the sequence directly and the FunOverlap weight was omitted; we label it the sequence variant. XGBoost follows Appel and Butcher (2026) on the same ESM-2 representation. Stacking fits one second-level logistic regression per term on IA (Wolpert 1992). The quantum neural network (QNN) has the architecture of PM-Net with θ_p fed to a 10-qubit, three-layer circuit, and the ZZ quantum kernel (Havlíček et al. 2019) is used in kernel ridge regression and compared with a classical RBF kernel.

Table 6. Compared methods.

Group	Method	Role
CAFA baseline	Naive	Predicts the training prior of each term
Literature: homology	DiamondScore	Bit-score-weighted transfer from DIAMOND hits
Literature: classical ML	GODoc (sequence variant)	kNN voting with score propagation; no PSSM, no FunOverlap
Literature: deep learning	DeepGOCNN, DeepGOPlus	CNN on one-hot sequence; CNN plus DiamondScore
Literature: deep learning	TALE, TALE+	Transformer with label embedding; plus DiamondScore
Literature: deep learning	ProLanGO	Neural machine translation, LSTM encoder–decoder
Classical on ESM-2	LR, Cos-kNN, MLP, RF, PM-Net	The five base models, each alone
Classical on ESM-2	XGBoost	Gradient boosting baseline (Appel and Butcher 2026)
Classical ensemble	Simple mean; LR stacking	Unweighted mean of the five models; per-term stacking on IA
Quantum	QNN	Matched control: PM-Net
Quantum	ZZ quantum kernel	Matched control: RBF kernel ridge regression
Proposed	QUBO-EASE	Per-term QUBO weights plus post-processing

4.4 Evaluation metrics

Protein-centric maximum F-measure. With T_i the true and $P_i(\tau)$ the predicted term set of protein i at threshold τ,

$$pr(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|}, \qquad rc(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{|P_i(\tau) \cap T_i|}{|T_i|}, \qquad F_{\max} = \max_\tau \frac{2\, pr(\tau)\, rc(\tau)}{pr(\tau) + rc(\tau)}, \tag{22}$$

where $m(\tau)$ is the number of proteins with at least one prediction at τ (Radivojac et al. 2013). The threshold is swept over 101 values from 0 to 1, and in each aspect only proteins with at least one positive label in that aspect are included. $F_{\max}$ is computed per aspect and the mean over the three aspects, $\bar{F}_{\max}$, is the summary criterion.

Complementary metrics. We also report the information-weighted $w\bar{F}_{\max}$ and the semantic distance

$$S_{\min} = \min_\tau \sqrt{ru(\tau)^2 + mi(\tau)^2}, \tag{23}$$

with remaining uncertainty ru and misinformation mi (Clark and Radivojac 2013). In this study the weight of a term in both metrics is its marginal information content, $\min\left(-\log_2 \pi_j, 25\right)$, which ignores the parent–child structure. The $S_{\min}$ values reported here are therefore not directly comparable with official CAFA values.

Statistical protocol. To separate observed differences from the variability due to the sample of test proteins we use the paired bootstrap (Efron and Tibshirani 1993). In each of 200 resamples the 6,000 test proteins are drawn with replacement and the difference in $\bar{F}_{\max}$ between two methods is computed on the same sample. A difference is called significant only when its 95% percentile interval excludes zero. Every method is run with one seed, so the variability due to training is not captured by this test.

4.5 Quantitative comparison

Table 7 reports all methods on the test split and Figure 3 orders them by $\bar{F}_{\max}$. QUBO-EASE reaches 0.5592, the highest value in the table, followed by LR stacking at 0.5479. The best literature reimplementation is DeepGOPlus at 0.5236. QUBO-EASE also has the highest $w\bar{F}_{\max}$ (0.5363). It is not the best method on $S_{\min}$, where LR stacking (10.895) and the simple mean (10.981) are better than its 11.203.

Table 7. All methods on the CAFA6 test split (one seed). F̄_max and wF̄_max are aspect means; S_min is computed over all terms (lower is better). The time of QUBO-EASE includes the training of the five base models.

Method	Group	F̄_max	MF	BP	CC	wF̄_max	S_min	Time (s)
Naive	CAFA baseline	0.3687	0.7060	0.1092	0.2909	0.3424	12.814	0
DiamondScore	Literature	0.4651	0.6061	0.3453	0.4440	0.4411	11.868	48
GODoc (sequence variant)	Literature	0.4045	0.6907	0.1765	0.3462	0.3753	12.676	39
DeepGOCNN	Literature	0.4618	0.7120	0.2209	0.4524	0.4353	11.850	1879
DeepGOPlus	Literature	0.5236	0.7278	0.3493	0.4936	0.5000	11.622	1927
TALE †	Literature	0.3701	0.7060	0.1133	0.2909	0.3436	12.717	311
TALE+ †	Literature	0.5080	0.7327	0.3449	0.4463	0.4837	11.840	979
ProLanGO ‡	Literature	0.3854	0.6959	0.1148	0.3454	0.3560	12.625	557

Method	Group	F̄_max	MF	BP	CC	wF̄_max	S_min	Time (s)
LR	Classical on ESM-2	0.4512	0.5679	0.3085	0.4773	0.4231	11.889	15
Cos-kNN	Classical on ESM-2	0.5080	0.7142	0.3058	0.5040	0.4822	11.240	0
MLP	Classical on ESM-2	0.4698	0.5591	0.3404	0.5098	0.4338	11.680	13
RF	Classical on ESM-2	0.4600	0.7179	0.2301	0.4321	0.4346	11.847	66
PM-Net	Classical on ESM-2	0.4533	0.5085	0.3519	0.4994	0.4189	11.633	17
XGBoost	Classical on ESM-2	0.5252	0.7223	0.3250	0.5283	0.5010	11.222	262
Simple mean	Classical ensemble	0.5280	0.6722	**0.3683**	0.5434	0.5025	10.981	111
LR stacking	Classical ensemble	0.5479	0.7335	0.3556	0.5545	0.5243	**10.895**	112
QNN	Quantum	0.4574	0.5181	0.3523	0.5020	0.4242	11.659	927
ZZ quantum kernel	Quantum	0.4419	0.7127	0.2254	0.3876	0.4164	12.015	19
QUBO-EASE	Proposed	**0.5592**	**0.7438**	0.3631	**0.5707**	**0.5363**	11.203	513

† Training of the TALE reimplementation did not succeed (see text). ‡ ProLanGO reached a score close to the naive baseline. Bold marks the best value in each column.

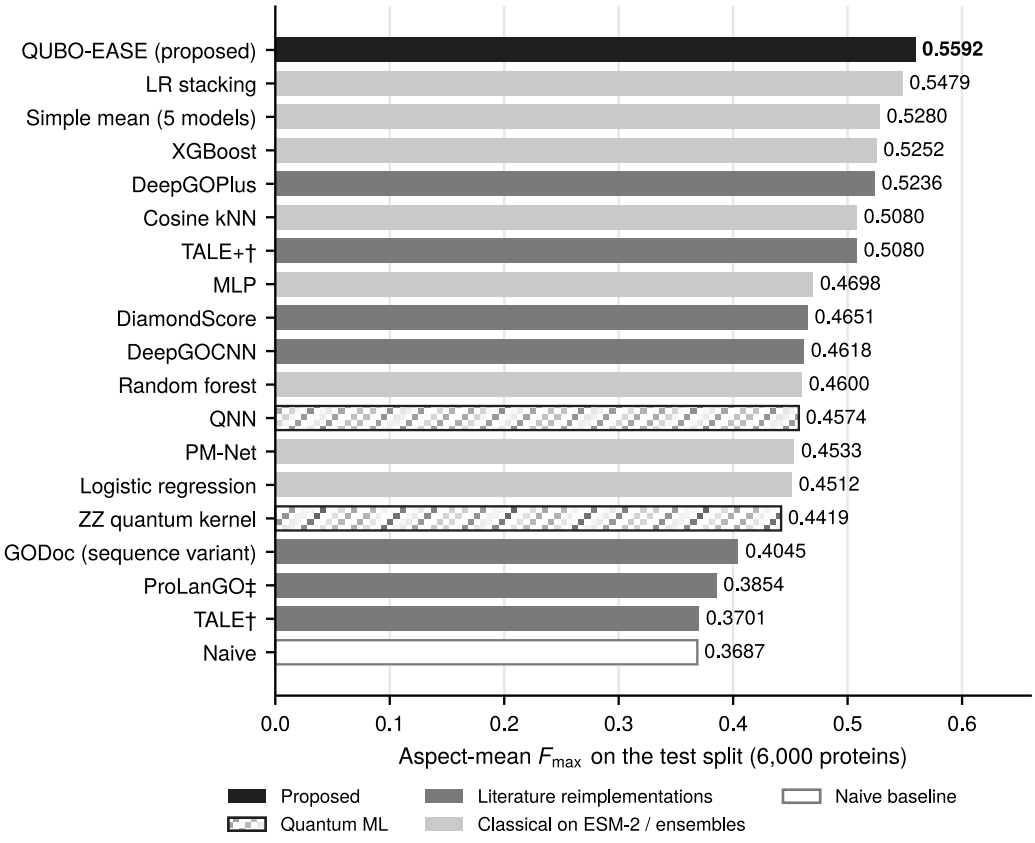


Fig 3. Aspect-mean F_max of every method on the test split (6,000 proteins, one seed).

Table 8 gives the paired differences between QUBO-EASE and each other method, and Figure 4 shows the differences for which interval bounds are tabulated. The smallest difference is against LR stacking (0.0116) and the largest against the naive baseline (0.1907); in the executed run none of the 18 intervals crossed zero.

Table 8. Paired bootstrap difference in F̄_max, QUBO-EASE minus each method (200 resamples). Differences are bootstrap means and can differ from the difference of the Table 7 entries in the last digit.

Method	F̄_max	Difference	95% interval	Outcome
LR stacking	0.5479	+0.0116	excludes zero §	significant
Simple mean	0.5280	+0.0309	excludes zero §	significant
XGBoost	0.5252	+0.0344	excludes zero §	significant
DeepGOPlus	0.5236	+0.0355	[+0.0284, +0.0423]	significant

Method	F̄_max	Difference	95% interval	Outcome
TALE+ †	0.5080	+0.0512	[+0.0444, +0.0576]	significant
Cos-kNN	0.5080	+0.0520	excludes zero §	significant
MLP	0.4698	+0.0897	excludes zero §	significant
DiamondScore	0.4651	+0.0936	[+0.0856, +0.1000]	significant
DeepGOCNN	0.4618	+0.0973	[+0.0902, +0.1045]	significant
RF	0.4600	+0.0991	excludes zero §	significant
QNN	0.4574	+0.1016	excludes zero §	significant
PM-Net	0.4533	+0.1059	excludes zero §	significant
LR	0.4512	+0.1066	excludes zero §	significant
ZZ quantum kernel	0.4419	+0.1168	excludes zero §	significant
GODoc (sequence variant)	0.4045	+0.1547	[+0.1459, +0.1616]	significant
ProLanGO ‡	0.3854	+0.1742	[+0.1661, +0.1814]	significant
TALE †	0.3701	+0.1893	[+0.1821, +0.1964]	significant
Naive	0.3687	+0.1907	excludes zero §	significant

§ The interval excluded zero in the executed run; its numerical bounds are not tabulated in the source thesis and are to be inserted from the notebook output before submission.

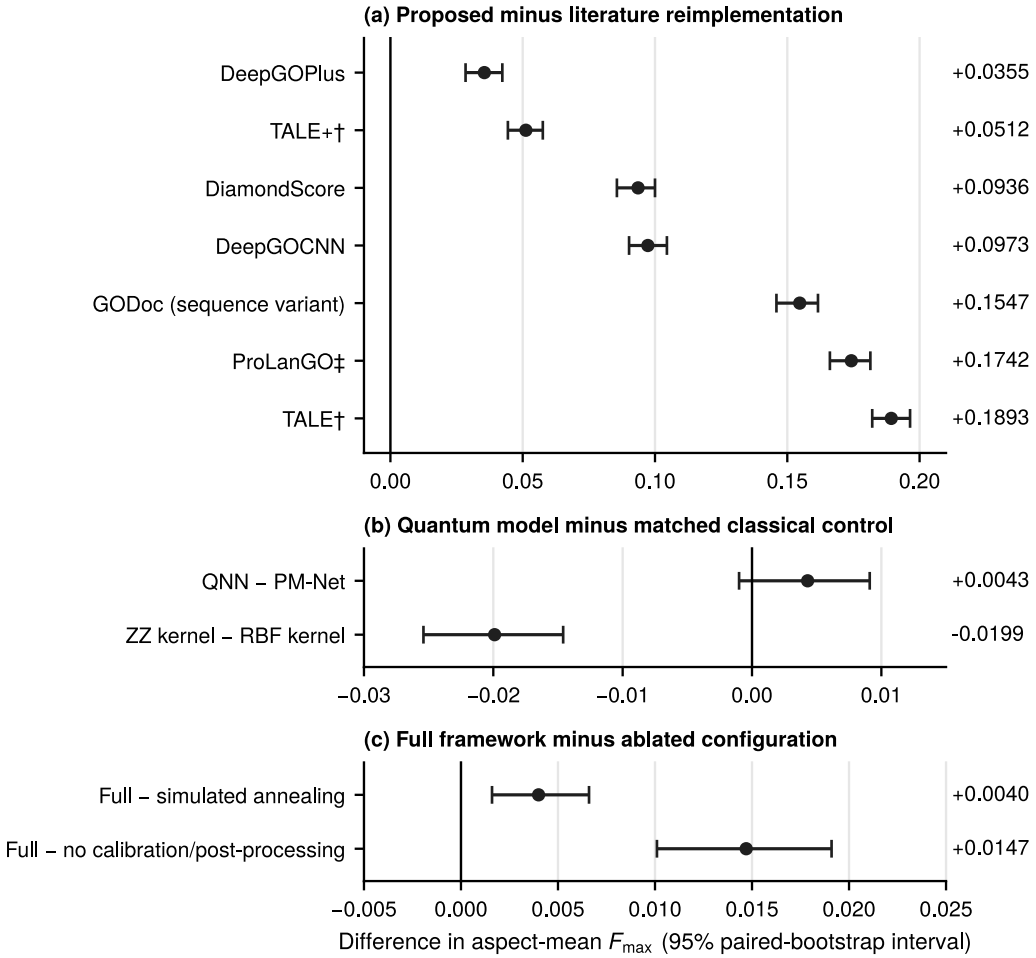


Fig 4. Paired bootstrap differences in aspect-mean F_max with 95% intervals. (a) QUBO-EASE minus each literature reimplementation. (b) Each quantum model minus its matched classical control. (c) Full framework minus each ablated configuration.

Literature reimplementations. The internal pattern of DeepGOPlus agrees with the original report: neither component is strong alone (DiamondScore 0.4651, DeepGOCNN 0.4618), but their combination with selected weights 0.3, 0.6 and 0.3 for the three aspects reaches 0.5236 (Kulmanov and Hoehndorf 2020). DiamondScore is below the

naive baseline in MF (0.6061 against 0.7060); one reason is that some test proteins had no hit with a sufficient E-value and kept a score of zero. DeepGOCNN overfitted after the fourth epoch and early stopping selected that epoch.

The TALE and ProLanGO rows need caution. The $F_{\max}$ of the single TALE model equals the naive baseline exactly in MF and CC (0.7060 and 0.2909), and its validation $F_{\max}$ did not exceed about 0.37 during training. The model did not move beyond predicting base frequencies, so its training failed in this reimplementation. The most likely cause is the training configuration (a learning rate without warm-up and early stopping with short patience); we did not test this explanation. The TALE row must not be read as the performance of the original method, and the TALE+ result stems mainly from its DiamondScore component. ProLanGO at 0.3854 is also close to the naive baseline, which is consistent with the ranking weakness reported in the original paper (Cao et al. 2017), but under-training cannot be excluded. The GODoc row reflects the approximate sequence variant and not the original method.

Quantum models against matched controls. The QNN scored 0.4574 against 0.4533 for PM-Net; the difference of +0.0043 has the interval [−0.0010, +0.0091], which contains zero, and its training took 927 s, about 55 times that of the control. The ZZ quantum kernel scored 0.4419 against 0.4620 for the RBF kernel, a difference of −0.0199 with interval [−0.0254, −0.0146]. Both quantum models are far below QUBO-EASE (by 0.1016 and 0.1168). These results agree with earlier findings that standalone quantum models show no significant advantage over matched controls on this task (Majidi 2026a; Majidi 2026b; Bowles et al. 2024).

4.6 Ablation study

Table 9 compares the full framework with two configurations. In the first, the same instances are solved by simulated annealing in place of exhaustive search and every other step is unchanged; this measures the effect of solution quality. In the second, the debiased channel for MF and the boosted channel for BP and CC, computed with the same QUBO weights, are used as the final prediction without calibration, EASE, prior or normalisation; this measures the contribution of the whole post-processing stage.

Table 9. Ablation on the test split. The last column is the full framework minus the configuration, with the 95% paired bootstrap interval.

Configuration	F̄_max	MF	BP	CC	Difference (95% interval)
QUBO-EASE (full)	0.5592	0.7438	0.3631	0.5707	—
Solved by simulated annealing	0.5556	0.7423	0.3547	0.5698	+0.0040 [+0.0016, +0.0066]
Without calibration and post-processing	0.5449	0.7372	0.3710	0.5265	+0.0147 [+0.0101, +0.0191]

Removing calibration and post-processing lowers $\bar{F}_{\max}$ by 0.0147, and the interval excludes zero. The largest drop is in CC, from 0.5707 to 0.5265, whereas in BP the configuration without post-processing is slightly higher (0.3710 against 0.3631). Solving by simulated annealing lowers $\bar{F}_{\max}$ by 0.0040, which is also significant, so solution quality has a small effect. Part of this difference may come from the λ_0 term, which the annealing solver omits.

Two contributions were not isolated. The independent contribution of EASE was not tested: its direct share selected on IB is non-zero only in BP (0.28), although the sharp EASE channel acts in all aspects through the calibrator. The independent contribution of the QUBO weights was not tested either: LR stacking and the simple mean do not receive the post-processing stage, so the differences of 0.0116 and 0.0309 against them do not separate the weighting from the post-processing.

4.7 Complexity analysis

Table 10 gives the time complexity of each stage, with N training proteins, N' query proteins, $n = 1280$ features, C terms, M models, K bits, I optimisation steps, hidden width h, B trees and Λ QUBO hyperparameter configurations.

Table 10. Time complexity of the stages of the framework.

Stage	Time complexity	Remark
Logistic regression	$O(I\,N\,n\,C)$	One linear layer
Cosine nearest neighbours	$O(N'N\,n)$	Matrix product on GPU
MLP and PM-Net	$O\big(I\,N\,h\,(n+C)\big)$	$h = 512$
Random forest	$O\big(B\sqrt{n}\,N\log N\big)$	On a 12,000-protein subsample
IA statistics	$O(M^2N'C)$	Includes pairwise synergy and leave-one-out
Exhaustive QUBO search	$O(\Lambda\,C\,2^{MK}M^2)$	Vectorised over the terms of an aspect
EASE fit	$O(NC^2 + C^3)$	Negligible for $C = 180$

Stage	Time complexity	Remark
Calibration and post-processing	$O(N'C^2 + \Lambda'N'C)$	Λ' post-processing configurations

The cost of exhaustive search grows exponentially with MK. Here each aspect requires 432 configurations, 60 terms and 32,768 states, about 850 million energy evaluations, which vectorised computation completes in a few minutes; the full pipeline took 513 s including base-model training (Table 7). With 10 models and 4 bits the search space reaches 2^{40} and exhaustive search is no longer feasible. That is the regime in which approximate classical and quantum solvers become relevant. The instances of different terms are independent and can be solved in parallel.

4.8 Visualization analysis

Figure 5 compares ten selected methods per aspect. In MF, QUBO-EASE reaches 0.7438 and the best other method, LR stacking, 0.7335. In this aspect the naive baseline at 0.7060 is above several learned models, including LR (0.5679) and the QNN (0.5181), which shows that a large part of the MF score comes from frequent terms. In BP, QUBO-EASE reaches 0.3631 while the simple mean of the five models reaches 0.3683, so the proposed framework is not the best method in BP. In CC, QUBO-EASE has the highest value, 0.5707 against 0.5545 for LR stacking.

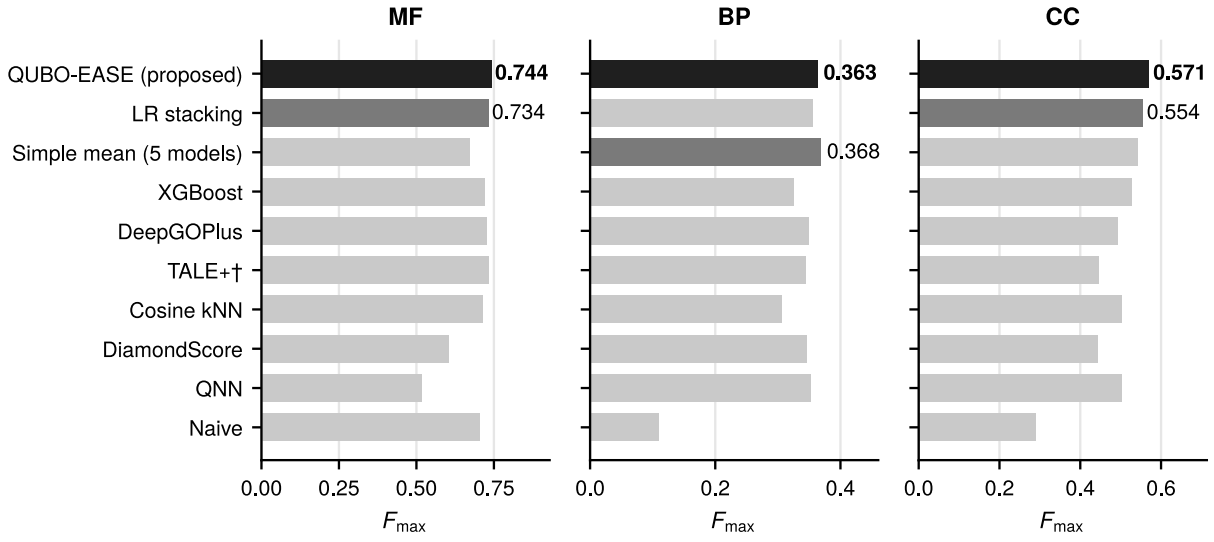


Fig 5. F_max per aspect for ten selected methods. Dark bars: QUBO-EASE; mid-grey bars: the best other method in each aspect.

Table 11 and Figure 6 give the mean normalised QUBO weights. The framework selects a different combination in each aspect. In MF the largest weight goes to logistic regression (0.402). In BP and CC the main weight is shared between the MLP and PM-Net, and the random forest is dropped. Cos-kNN, MLP and PM-Net are selected for every term, logistic regression for about half of the terms and the random forest for one third.

Table 11. Mean normalised weight of each base model in the QUBO weighting, per aspect, and the fraction of all 180 terms for which the model is selected.

Base model	MF	BP	CC	Selection rate
LR	0.402	0.004	0.015	0.528
Cos-kNN	0.126	0.222	0.055	1.000
MLP	0.259	0.388	0.455	1.000
RF	0.164	0.000	0.000	0.333
PM-Net	0.049	0.386	0.475	1.000

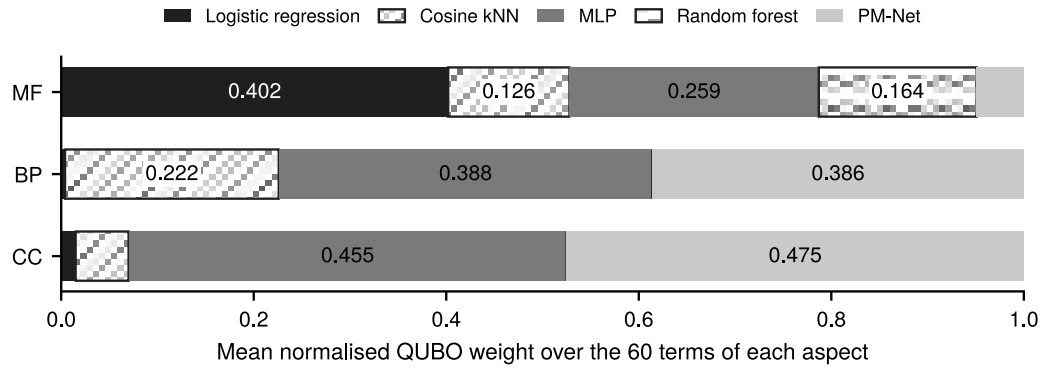


Fig 6. Share of the mean normalised QUBO weight of each base model in each aspect (mean over the 60 terms of the aspect).

5. Discussion and Conclusion

The results support four statements. First, QUBO-EASE has the highest aspect-mean $F_{\max}$ of the 19 compared methods, 0.5592, and its difference from each of the other 18, including DeepGOPlus, XGBoost, classical stacking and both quantum models, was significant under the paired bootstrap. Second, this advantage belongs to the pipeline as a whole: complementary base models, their combination, calibration and post-processing. The ablation shows that removing the post-processing stage costs 0.0147, and the independent contribution of the QUBO weighting was not separated. Third, neither standalone quantum model was significantly better than its matched classical control. Fourth, among the literature methods the combined systems, DeepGOPlus and TALE+, were better than their single components, which is the same logic of complementary models on which the framework rests.

The advantage is not uniform. In BP the simple mean is higher than the proposed framework, and the configuration without post-processing is higher still (0.3710); the post-processing coefficients selected on IB did not transfer to the test split in this aspect. On $S_{\min}$ both classical ensembles are better. The weights in Table 11 indicate why per-term weighting can help: the useful models differ between aspects, and a single weight vector cannot express that.

The word “quantum” needs a precise reading. The instances were solved by classical exhaustive search, and no claim of quantum advantage is made. What the study contributes on the quantum side is a formulation: per-term ensemble weighting fits in 15 binary variables, within the reach of current devices (Preskill 2018), and it does not require compressing the protein representation. Whether a quantum annealer or QAOA can solve larger instances of this problem better or faster than simulated annealing is an open question that the present experiments do not address. The significant loss of 0.0040 under simulated annealing indicates that solution quality matters for the final score, so that question is worth asking.

Table 12 states the status of the four hypotheses that guided the study. We mark the main hypothesis as partially supported because the improvement was obtained but is not attributable to the quantum-compatible component, and because all results come from one seed and one random split.

Table 12. Status of the research hypotheses.

Hypothesis	Main evidence	Status
H: the framework improves F̄_max without dimensionality reduction	Significant difference from all 18 methods, e.g. +0.0355 against DeepGOPlus and +0.0344 against XGBoost	Partially supported
H1: QUBO weighting beats single models and classical ensembles	+0.0116 against stacking, +0.0309 against the simple mean; post-processing not matched	Partially supported
H2: EASE with prior and normalisation mitigates sparsity	Independent contribution of EASE not isolated	Not tested
H3: removing QUBO or EASE lowers performance	−0.0147 without calibration and post-processing; separate removals not run	Partially supported

Several constraints bound what the study establishes. All results come from one random seed, and the bootstrap intervals reflect only the sampling of test proteins, not the variability of training. The split is random, so homologous proteins can occur in training and test data and performance on novel proteins may be lower. The label space is limited to 180 frequent terms. The hyperparameter search space of the post-processing stage of the proposed framework was larger than that of the other methods, which is an advantage in tuning that the comparison does not correct for. The literature reimplementations are approximate: GODoc ran without PSSM and FunOverlap, TALE training failed, ProLanGO stayed near the naive baseline, and the TALE hyperparameters were not searched as in the original paper;

the figures here are therefore not comparable with the published CAFA3 results of these methods. The $S_{\min}$ values use marginal information content.

The next experiments follow from these limits, in order of priority: (i) repeat the study over several seeds and on a cluster-disjoint split by sequence similarity (Steinegger and Söding 2017), and on a temporal split following the CAFA protocol; (ii) run separate ablations for EASE and for the QUBO weights, including stacking and the simple mean with the same post-processing stage, to isolate each contribution; (iii) solve the instances on quantum hardware or with QAOA and compare solution quality and time with exhaustive search and simulated annealing; (iv) extend the label space to rare terms, where solving one instance per term becomes costly and grouped or hierarchical formulations may be needed; (v) add homology-based and structure-based base models, given that DeepGOPlus benefits from homology; (vi) rerun TALE with a corrected training configuration and GODoc with PSSM; and (vii) add a term to the energy that enforces consistency of the weights between parent and child terms.

In conclusion, a quantum-compatible formulation can be placed at a stage of the protein function prediction pipeline that respects the qubit budget and does not discard information from the representation. On 40,000 CAFA6 proteins and 180 GO terms the resulting framework reached an aspect-mean $F_{\max}$ of 0.5592 and scored above all 18 compared methods, while the standalone quantum models showed no significant advantage over their classical controls. The present value of the approach lies in pipeline design and in combining complementary models; the actual contribution of quantum computation to it remains to be determined by the experiments listed above.

Statements and Declarations

-Acknowledgements

This article is derived from the first author's thesis in Computer Engineering (Software) at Islamic Azad University, Sanandaj Branch, supervised by the second author. The authors thank Dr. Isa Mohammadi, who advised the thesis.

-Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

-Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

-Data Availability

The CAFA 6 data used in this study are publicly available from the challenge organisers. The list of the 180 selected GO terms, the split indices and the result files from which every table was built are available from the corresponding author upon reasonable request.

-Code Availability

The executed Kaggle notebook that produces every reported result is available from the corresponding author upon reasonable request.

-Author Contributions

F. Majidi conceived and designed the study, implemented the models and the evaluation pipeline, performed the experiments and analysis, and wrote the manuscript. A. Bahrampour supervised the research and reviewed the manuscript. Both authors approved the final manuscript.

-Ethics Approval

Not applicable. This study uses only publicly available protein sequence and annotation data and involves no human participants or animals.

-Consent to Participate and Consent for Publication

Not applicable.

-Use of AI Tools

An AI language model (Claude, Anthropic) assisted with drafting the English manuscript from the first author's thesis and with reference formatting. All reported numbers derive from the executed notebook. The authors are responsible for reviewing the final manuscript and verifying all claims and citations.

References

S. F. Altschul et al. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Research, 25, 17, 3389–3402, https://doi.org/10.1093/nar/25.17.3389.

J. Appel and N. Butcher (2026) Protein function prediction with pretrained ProtT5 embeddings and gradient boosting. bioRxiv, https://doi.org/10.64898/2026.04.27.721184.

M. Ashburner et al. (2000) Gene ontology: Tool for the unification of biology. Nature Genetics, 25, 1, 25–29, https://doi.org/10.1038/75556.

J. Bowles, S. Ahmed, and M. Schuld (2024) Better than classical? The subtle art of benchmarking quantum machine learning models. arXiv:2403.07059.

L. Breiman (2001) Random forests. Machine Learning, 45, 1, 5–32, https://doi.org/10.1023/A:1010933404324.

B. Buchfink, C. Xie, and D. H. Huson (2015) Fast and sensitive protein alignment using DIAMOND. Nature Methods, 12, 1, 59–60, https://doi.org/10.1038/nmeth.3176.

R. Cao et al. (2017) ProLanGO: Protein function prediction using neural machine translation based on a recurrent neural network. Molecules, 22, 10, article 1732.

Y. Cao and Y. Shen (2021) TALE: Transformer-based protein function annotation with joint sequence–label embedding. Bioinformatics, 37, 18, 2825–2833, https://doi.org/10.1093/bioinformatics/btab198.

R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes (2004) Ensemble selection from libraries of models. in Proc. 21st International Conference on Machine Learning, 18, https://doi.org/10.1145/1015330.1015432.

W. T. Clark and P. Radivojac (2013) Information-theoretic evaluation of predicted ontological annotations. Bioinformatics, 29, 13, i53–i61, https://doi.org/10.1093/bioinformatics/btt228.

B. Efron and R. J. Tibshirani (1993) An Introduction to the Bootstrap. Chapman & Hall, New York.

E. Farhi, J. Goldstone, and S. Gutmann (2014) A quantum approximate optimization algorithm. arXiv:1411.4028.

Gene Ontology Consortium (2023) The Gene Ontology knowledgebase in 2023. Genetics, 224, 1, iyad031, https://doi.org/10.1093/genetics/iyad031.

F. Glover, G. Kochenberger, and Y. Du (2019) Quantum Bridge Analytics I: A tutorial on formulating and using QUBO models. 4OR, 17, 335–371, https://doi.org/10.1007/s10288-019-00424-y.

V. Havlíček et al. (2019) Supervised learning with quantum-enhanced feature spaces. Nature, 567, 209–212, https://doi.org/10.1038/s41586-019-0980-2.

G. E. Hinton (2002) Training products of experts by minimizing contrastive divergence. Neural Computation, 14, 8, 1771–1800, https://doi.org/10.1162/089976602760128018.

Iowa State University et al. (2025) CAFA 6 protein function prediction. Kaggle, https://www.kaggle.com/competitions/cafa-6-protein-function-prediction.

Y. Jiang et al. (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biology, 17, article 184, https://doi.org/10.1186/s13059-016-1037-6.

T. Kadowaki and H. Nishimori (1998) Quantum annealing in the transverse Ising model. Physical Review E, 58, 5, 5355–5363, https://doi.org/10.1103/PhysRevE.58.5355.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi (1983) Optimization by simulated annealing. Science, 220, 4598, 671–680, https://doi.org/10.1126/science.220.4598.671.

A. Krogh and J. Vedelsby (1995) Neural network ensembles, cross validation, and active learning. Advances in Neural Information Processing Systems, 7, 231–238.

M. Kulmanov and R. Hoehndorf (2020) DeepGOPlus: Improved protein function prediction from sequence. Bioinformatics, 36, 2, 422–429, https://doi.org/10.1093/bioinformatics/btz595.

M. Kulmanov, M. A. Khan, and R. Hoehndorf (2018) DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier. Bioinformatics, 34, 4, 660–668, https://doi.org/10.1093/bioinformatics/btx624.

M. Kulmanov et al. (2024) Protein function prediction as approximate semantic entailment. Nature Machine Intelligence, 6, 220–228, https://doi.org/10.1038/s42256-024-00795-w.

R. Y. Li et al. (2018) Quantum annealing versus classical machine learning applied to a simplified computational biology problem. npj Quantum Information, 4, article 14, https://doi.org/10.1038/s41534-018-0060-8.

Z. Lin et al. (2023) Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 379, 6637, 1123–1130, https://doi.org/10.1126/science.ade2574.

Y.-W. Liu et al. (2020) GODoc: High-throughput protein function prediction using novel k-nearest-neighbor and voting algorithms. BMC Bioinformatics, 21, suppl. 6, article 276, https://doi.org/10.1186/s12859-020-03556-9.

I. Loshchilov and F. Hutter (2019) Decoupled weight decay regularization. in Proc. 7th International Conference on Learning Representations (ICLR), arXiv:1711.05101.

A. Lucas (2014) Ising formulations of many NP problems. Frontiers in Physics, 2, article 5, https://doi.org/10.3389/fphy.2014.00005.

F. Majidi (2026a) A dimension-matched quantum feature map for hierarchical Gene Ontology term prediction: Attributing the gain on CAFA6. Research Square, preprint, https://doi.org/10.21203/rs.3.rs-10501636/v1.

F. Majidi (2026b) Matched-control benchmarking of quantum machine learning models for hierarchical protein function prediction. Research Square, preprint, https://doi.org/10.21203/rs.3.rs-11188049/v1.

J. R. McClean et al. (2018) Barren plateaus in quantum neural network training landscapes. Nature Communications, 9, article 4812, https://doi.org/10.1038/s41467-018-07090-4.

A. Mott et al. (2017) Solving a Higgs optimization problem with quantum annealing for machine learning. Nature, 550, 375–379, https://doi.org/10.1038/nature24047.

H. Neven et al. (2008) Training a binary classifier with the quantum adiabatic algorithm. arXiv:0811.0416.

H. Neven et al. (2012) QBoost: Large scale classifier training with adiabatic quantum optimization. in Proc. Asian Conference on Machine Learning, PMLR, 25, 333–348.

A. Paszke et al. (2019) PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, 8024–8035.

F. Pedregosa et al. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

J. C. Platt (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. in Advances in Large Margin Classifiers, MIT Press, 61–74.

J. Preskill (2018) Quantum computing in the NISQ era and beyond. Quantum, 2, 79, https://doi.org/10.22331/q-2018-08-06-79.

P. Radivojac et al. (2013) A large-scale evaluation of computational protein function prediction. Nature Methods, 10, 3, 221–227, https://doi.org/10.1038/nmeth.2340.

A. Rives et al. (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences, 118, 15, e2016239118, https://doi.org/10.1073/pnas.2016239118.

M. Schuld and N. Killoran (2019) Quantum machine learning in feature Hilbert spaces. Physical Review Letters, 122, 4, 040504, https://doi.org/10.1103/PhysRevLett.122.040504.

H. Steck (2019) Embarrassingly shallow autoencoders for sparse data. in Proc. The World Wide Web Conference (WWW '19), 3251–3257, https://doi.org/10.1145/3308558.3313710.

M. Steinegger and J. Söding (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nature Biotechnology, 35, 11, 1026–1028, https://doi.org/10.1038/nbt.3988.

S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes (2024) Exponential concentration in quantum kernel methods. Nature Communications, 15, article 5200, https://doi.org/10.1038/s41467-024-49287-w.

S. Wang et al. (2023) NetGO 3.0: Protein language model improves large-scale functional annotations. Genomics, Proteomics & Bioinformatics, 21, 2, 349–358, https://doi.org/10.1016/j.gpb.2023.04.001.

D. H. Wolpert (1992) Stacked generalization. Neural Networks, 5, 2, 241–259, https://doi.org/10.1016/S0893-6080(05)80023-1.

R. You et al. (2018) GOLabeler: Improving sequence-based large-scale protein function prediction by learning to rank. Bioinformatics, 34, 14, 2465–2473, https://doi.org/10.1093/bioinformatics/bty130.

R. You et al. (2019) NetGO: Improving large-scale protein function prediction with massive network information. Nucleic Acids Research, 47, W1, W379–W387, https://doi.org/10.1093/nar/gkz388.

Q. Yuan et al. (2023) Fast and accurate protein function prediction from sequence through pretrained language model and homology-based label diffusion. Briefings in Bioinformatics, 24, 3, bbad117, https://doi.org/10.1093/bib/bbad117.

N. Zhou et al. (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biology, 20, article 244, https://doi.org/10.1186/s13059-019-1835-8.