Unlocking Noisy Real-World Corpora for Foundation Model Pre-Training via Quality-Aware Tokenization

Arvid E. Gollwitzer [1 2 3] **David de Gruijl** [3] **Paridhi Latawa** [2] **Deepak A. Subramanian** [1 4]
Adrián Noriega de la Colina [1 2 5 6]

Abstract

Current tokenization methods process sequential data without accounting for signal quality, limiting their effectiveness on noisy real-world corpora. We present *QA-Token (Quality-Aware Tokenization)*, which incorporates data reliability directly into vocabulary construction. We make three key contributions: (i) a bilevel optimization formulation that jointly optimizes vocabulary construction and downstream performance, (ii) a reinforcement learning approach that learns merge policies through quality-aware rewards with convergence guarantees, and (iii) an adaptive parameter learning mechanism via Gumbel-Softmax relaxation for end-to-end optimization. Our experimental evaluation demonstrates consistent improvements: *genomics* (6.7 percentage point F1 gain in variant calling over BPE), *finance* (30% Sharpe ratio improvement). At foundation scale, we re-tokenize the METAGENE-1 pretraining corpus comprising 1.7 trillion base-pairs and achieve state-of-the-art pathogen detection (94.53 MCC) while reducing token count by 15%. We unlock noisy real-world corpora, spanning petabases of genomic sequences and terabytes of financial time series, for foundation model training with zero inference overhead.

1. Introduction

Tokenization serves as the interface between raw data and neural computation. Current methods such as Byte-Pair Encoding (BPE) (Sennrich et al., 2016) rely exclusively on frequency statistics, assuming that occurrence frequency correlates with semantic importance. This assumption fails when data quality varies significantly—from sequencing errors in genomics (Ewing et al., 1998) to microstructure noise in financial markets (Andersen et al., 2001). Models trained on noisy corpora using frequency-based tokenization inherit these errors, resulting in degraded performance—an effect now formalized through quality-aware scaling laws (Subramanyam et al., 2025).

The problem is substantial: error rates in third-generation sequencing exceed 10% (Wenger et al., 2019), yet current tokenizers treat high-confidence and error-prone regions identically. In finance, over 40% of high-frequency data contains microstructure noise (Hansen & Lunde, 2006), but tokenization methods do not distinguish signal quality. This limitation constrains foundation model training on real-world data.

The scale of available biological data amplifies this challenge. Public sequence repositories now contain over 67 petabase pairs (Pbp) of raw sequencing data, with the European Nucleotide Archive doubling approximately every 45 months (Karasikov et al., 2025). Recent advances in efficient indexing have made these petabase-scale archives full-text searchable at costs as low as $0.74 per queried megabase pair, demonstrating that the infrastructure for large-scale sequence analysis is maturing rapidly. However, a substantial fraction of this data remains underutilized for foundation model training due to quality heterogeneity—standard frequency-based tokenization methods either discard low-quality reads entirely or propagate sequencing errors into learned representations. This gap between data availability and usability motivates a fundamental rethinking of how tokenization handles quality variation.

We present **Quality-Aware Tokenization (QA-Token)**, a framework that incorporates data quality into vocabulary construction. We make three key contributions:

1. Bilevel Optimization with Complexity Analysis: We formalize tokenization as a bilevel optimization problem (Theorem 3.1) that jointly optimizes vocabulary construction and downstream performance. We show this problem

[1]Broad Institute of MIT and Harvard, Cambridge, MA, USA [2]Massachusetts Institute of Technology, Cambridge, MA, USA [3]Anto Biosciences (YC F25) [4]Koch Institute for Integrative Cancer Research, MIT, Cambridge, MA, USA [5]Department of Neurology and Neurosurgery, McGill University, Montreal, Canada [6]The Montreal Neurological Hospital-Institute, Montreal, Canada. Correspondence to: Arvid E. Gollwitzer <arvidg@mit.edu>.

Preprint. February 10, 2026.

is NP-hard (Theorem 3.2) and develop a principled approximation scheme with theoretical guarantees.

2. Reinforcement Learning with Convergence Guarantees: We cast vocabulary construction as a Markov Decision Process (Theorem E.4) and employ reinforcement learning to discover optimal merge policies. We provide formal convergence analysis (Theorem E.5) and achieve $(1 - 1/e)$-approximation to the optimal adaptive policy.

3. Differentiable Parameter Learning: Through Gumbel-Softmax relaxation (Theorem C.8), we enable end-to-end learning of quality sensitivity parameters, with proven consistency and bounded gradients (Theorem C.7).

We show that QA-Token achieves information-theoretic optimality under noisy conditions (Theorem C.13), providing formal justification for quality-aware tokenization. Our evaluation shows 30% higher Sharpe ratios in algorithmic trading, 6.7 percentage point improvement in genomic variant calling F1 (0.891 vs. 0.824 for BPE), and state-of-the-art performance when integrated into 7B-parameter foundation models.

Core Contributions: (i) We derive a quality-aware merge score (Theorem C.3) balancing frequency, quality, and domain constraints with learnable sensitivity α (Section C.2). (ii) We formulate vocabulary construction as an MDP (Theorem E.4, Section E.7) achieving $(1 - 1/e)$-approximation through adaptive submodularity. (iii) Gumbel-Softmax relaxation enables end-to-end parameter learning with $O(1/\sqrt{T})$ convergence rate (Theorem E.2, Section E.4). (iv) Domain-specific instantiations achieve state-of-the-art performance across 15+ benchmarks.

Our analysis shows that incorporating quality signals into tokenization enables training on noisy corpora where frequency-based methods fail, expanding the range of usable training data for foundation models with broader scientific and economic implications (Section 7.1).

2. Quality Metrics for Noisy Domains

Quality metrics must satisfy three formal properties to enable principled integration into the merge score: (i) *boundedness* ($q \in [0, 1]$) ensuring numerical stability, (ii) *Lipschitz continuity* enabling stable gradient computation during adaptive learning, and (iii) *monotonicity under noise injection* (higher noise yields lower quality) ensuring semantic consistency. We prove these properties hold for our domain-specific instantiations (Theorem C.1, Section C.1).

Genomics: We leverage Phred scores with position-adjusted decay: $q'_{s_j} = q_{s_j} \cdot \exp(-\beta_{\text{pos}} \cdot j/L)$, where β_{pos} is learned and L is read length. Token quality is aggregated via geometric mean $q_t = (\prod_{j=1}^{|t|} q'_{s_j})^{1/|t|}$ to ensure sensitivity to low-quality regions—a single unreliable base compromises the entire token (Eq. 13, Section D).

Finance: We combine four market microstructure dimensions: (i) liquidity q_{liq} (bid-ask spread, depth), (ii) signal quality q_{sig} (SNR of price changes), (iii) stability q_{stb} (volatility regime), and (iv) information content q_{info} (order flow informativeness). The composite score $q_t^{\text{finance}} = \sum_k w_k q_{k,t}$ uses learned weights; arithmetic mean aggregation reflects additive noise characteristics of financial data (Eq. 14, Section D).

With quality metrics defined, we now formalize how they integrate into the tokenization objective.

3. Mathematical Formulation of QA-Token

3.1. Notation and Setup

Let $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ represent a corpus comprising N sequences, where each sequence $S_k = (s_{k,1}, \ldots, s_{k,n_k})$ consists of elements drawn from a base alphabet Σ. Each atomic element $s_{k,i}$ is associated with a normalized quality score $q_{k,i} \in [0, 1]$ as defined in Section 2. The initial vocabulary is defined as $V_0 = \Sigma$. At any step k of the tokenization process, V_k denotes the current vocabulary. For any token $a \in V_k$, we denote its frequency in the corpus as $f(a)$, and for an adjacent pair (a, b), their co-occurrence frequency is $f(a, b)$. The length of a token t in atomic units is $|t|$. Let q_t be the aggregated scalar quality of token t, computed using domain-specific aggregation functions (see Section D).

3.2. Formal Problem Definition and Objective

We formalize tokenization as finding a tokenizer $\mathcal{T}$ that maximizes objective $\mathcal{J}$, balancing downstream task performance, vocabulary complexity, and data reliability. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ denote a corpus of N sequences sampled from an underlying data distribution $\mathcal{P}_{\text{data}}$, where each $S_k = (s_{k,1}, \ldots, s_{k,n_k})$ consists of elements from base alphabet Σ. A tokenizer $\mathcal{T} : \mathcal{S} \to \mathcal{Z}$ maps the corpus to segmentations $\mathcal{Z} = \{Z_1, \ldots, Z_N\}$ using vocabulary V.

Definition 3.1 (Bilevel Tokenization Problem)**.** The optimal quality-aware tokenization problem is formulated as the following bilevel optimization:

$$\max_{\mathcal{T} \in \mathcal{G}(K)} \mathcal{J}(\mathcal{T}) := \lambda_{\text{LM}} \mathcal{L}_{\text{LM}}(\mathcal{T}) - \lambda_{\text{comp}} \Phi(V) + \lambda_{\text{qual}} Q(V, \mathcal{Z}), \tag{1}$$

where the language model performance is:

$$\mathcal{L}_{\text{LM}}(\mathcal{T}) = \max_{\theta \in \Theta} \mathbb{E}_{\mathcal{D} \sim \mathcal{P}_{\text{data}}}[\log p_\theta(\mathcal{D}|\mathcal{T})], \tag{2}$$

and $\mathcal{G}(K) = \{\mathcal{T} : |V_{\mathcal{T}}| - |\Sigma| \leq K\}$ denotes the set of tokenizers reachable by at most K merge operations from base alphabet Σ, with Θ being the parameter space of the language model.

The objective $\mathcal{J}$ balances three components: (i) downstream performance $\mathcal{L}_{\text{LM}}(\mathcal{T})$ maximizing expected log-likelihood, (ii) complexity penalty $\Phi(V) = |V| \log |V| + \sum_{t \in V} |t| \cdot H(t)$ following MDL principles (Rissanen, 1978)—the first term penalizes vocabulary size (description length of token indices), while the second penalizes internal token complexity via the empirical entropy $H(t) = -\sum_{\sigma \in \Sigma} \frac{n_\sigma(t)}{|t|} \log \frac{n_\sigma(t)}{|t|}$ of atomic elements within token t (with $n_\sigma(t)$ the count of element σ; $H(t) = 0$ for single-element tokens), and (iii) reliability reward $Q(V, \mathcal{Z}) = \frac{1}{\sum_{k=1}^{N} |Z_k|} \sum_{k=1}^{N} \sum_{t \in Z_k} g(q_t)$ aggregating token qualities through concave function g.

The aggregator function g exhibits concavity to capture diminishing returns for merging high-quality constituents. Throughout this work, we employ $g(x) = (x + \epsilon_Q)^\alpha$ with $0 < \alpha < 1$ (strictly concave) and $\epsilon_Q = 10^{-8}$ for numerical stability. The boundary case $\alpha = 1$ yields linear aggregation, which is appropriate when quality contributions are additive rather than subject to diminishing returns.

Theorem 3.2 (Computational Complexity)**.** *The bilevel optimization problem in Eq. 1 is NP-hard in general (Dempe, 2020); indeed, polynomial bilevel programming is Σ_2^p-hard (Cen & Chi, 2023), placing it one level above NP in the polynomial hierarchy. The worst case requires $O(|\Sigma|^K \cdot K! \cdot N \cdot n \cdot |\Theta|)$ evaluations (proof in Section C.5).*

Given this computational intractability, we develop a principled approximation scheme combining greedy merge selection with reinforcement learning, as detailed in subsequent sections.

3.3. Quality-Aware Merge Score

We extend PMI-based tokenization by incorporating quality signals through the following result:

Theorem 3.3 (Quality-Aware Merge Score)**.** *The optimal greedy merge score maximizing the first-order approximation of the objective increment $\Delta\mathcal{J}$ is:*

$$w_{ab} = \frac{f(a,b)}{f(a)f(b) + \epsilon_f} \cdot (\bar{q}_{ab} + \epsilon_Q)^\alpha \cdot \psi(a,b) \qquad (3)$$

where $\bar{q}_{ab} = (q_a + q_b)/2$ averages constituent qualities, $\alpha \in (0, 1]$ controls quality sensitivity, and $\psi(a,b)$ encodes domain constraints. (Proof via first-order approximation in Section C.2.)

This score balances statistical association (PMI term), data reliability (quality term), and domain-specific requirements. Boundedness and Lipschitz continuity are proven in Theorem C.4 (Section C.5).

4. Learning Framework: RL and Adaptive Parameters

We cast vocabulary construction as a learning problem with two sequential stages. **Stage 1** (RL Policy Optimization) learns policy π_{θ_π} for merge selection using PPO with quality-aware rewards, keeping initial parameters $\theta_{\text{adapt}}^{(0)}$ fixed. **Stage 2** (Adaptive Parameter Learning) optimizes θ_{adapt} via Gumbel-Softmax relaxation for downstream performance, using *greedy simulation* with composite logits $\ell_{ab}(\theta_{\text{adapt}})$ rather than invoking the RL policy directly—the Stage 1 policy serves to initialize candidate merges and provide variance reduction baselines. Gradients $\nabla_{\theta_{\text{adapt}}} L_{\text{task}}$ flow through Gumbel-Softmax merge selection, enabling end-to-end learning (detailed in Section E, Algorithms 1–3).

4.1. Reinforcement Learning Formulation

Definition 4.1 (Tokenization MDP)**.** The vocabulary construction MDP is $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, T)$ where: states $s_t \in \mathcal{S}$ encode current vocabulary, merge candidates, and corpus statistics; actions $a_t \in \mathcal{A}_t$ select merge pairs; transitions $\mathcal{P}$ are deterministic vocabulary updates; rewards $\mathcal{R}$ are quality-aware (Section 4.2); $\gamma \in (0, 1]$ is the discount factor; T is the horizon (target vocabulary size minus base alphabet size). Complete specification in Section E.7.

The RL objective finds policy $\pi_{\theta_\pi} : \mathcal{S} \to \Delta(\mathcal{A})$ maximizing expected cumulative reward over T operations using PPO (Schulman et al., 2017), with global convergence guarantees following (Bhandari & Russo, 2021; Cen & Chi, 2023). Theorem E.5 (Section E.7) proves MDP well-formedness.

4.2. Reward Function Design

The multi-objective reward $R(a, b; \theta_{\text{adapt}}^{(0)}) = \sum_j \lambda_j \hat{R}_j(a, b)$ combines quality, information, complexity, and domain-specific components. Each raw reward R_j^{raw} is normalized using adaptive running statistics with exponential moving averages: $\mu_{j,t}^{\text{run}} = (1 - \beta_{\text{norm}})\mu_{j,t-1}^{\text{run}} + \beta_{\text{norm}} R_j^{\text{raw}}$, yielding $\hat{R}_j = (R_j^{\text{raw}} - \mu_{j,t-1}^{\text{run}})/(\sigma_{j,t-1}^{\text{run}} + \epsilon_R)$. This ensures bounded, scale-invariant rewards during non-stationary policy optimization (Theorem C.5, Section E.8).

4.3. Adaptive Learning of Tokenization Parameters

After RL optimization, we learn θ_{adapt} (quality sensitivity α, domain factors $\beta_{\text{pos}}/\beta_{\text{vol}}$, weights) minimizing $L_{\text{total}}(\theta_{\text{adapt}}) = L_{\text{task}}(\theta_{\text{adapt}}) + \lambda_{\text{reg}} \|\theta_{\text{adapt}}\|_2^2$ via Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017). Temperature annealing $\tau(t) = \tau_{\text{init}} \exp(-\beta_{\text{anneal}} t / T_{\text{anneal}})$ ensures convergence (Theorem C.7, Theorem E.2; Section E.9, Section E). The two-stage framework—RL with fixed $\theta_{\text{adapt}}^{(0)}$ then adaptive learning—culminates in greedy vocabulary construction using $w_{ab}(a, b; \theta_{\text{adapt}}^*)$ (Section E, Algorithms 1–

3).

4.4. Two-Timescale Convergence

The sequential optimization follows a two-timescale stochastic approximation: policy updates on fast timescale (learning rate $\eta_\pi^{(t)}$), adaptive parameters on slow timescale ($\eta_{\text{adapt}}^{(t)}$), with $\eta_\pi^{(t)}/\eta_{\text{adapt}}^{(t)} \to \infty$ as $t \to \infty$. Under assumptions (A1)–(A4), this converges to a local Nash equilibrium where θ_π^* maximizes $J(\pi; \theta_{\text{adapt}}^*)$ and θ_{adapt}^* minimizes $L_{\text{total}}(\theta_{\text{adapt}}; \pi^*)$. Quality bounds and initialization strategies for approaching global optima are detailed in Section E.

4.5. Theoretical Guarantees

Our framework provides the following guarantees under assumptions (A1)–(A4) detailed in Section C.6: (i) bounded/Lipschitz merge scores w_{ab} (Theorem C.4), (ii) stable EMA normalization with strictly positive running standard deviations (Theorem C.5), (iii) PPO convergence to stationary points (Theorem C.6), (iv) consistent and bounded Gumbel-Softmax gradients (Theorem C.7), and (v) $(1 - 1/e)$-approximation to optimal adaptive policy via adaptive submodularity.

Information-Theoretic Optimality: Building on information bottleneck theory (Tishby et al., 1999; Alemi et al., 2017), our analysis (Theorem C.13, Section C.9) shows QA-Token minimizes the quality-aware information bottleneck: $\mathcal{L}_{\text{QA}}(V) = -I(T;Y|Q) + \beta \cdot I(T;X|Q)$, achieving optimal compression-relevance tradeoffs under noisy conditions by maximizing task-relevant information $I(T;Y|Q)$ while minimizing redundant representation complexity $I(T;X|Q)$, conditioned on quality Q. Complete proofs in Section C.5 and Section E.

Having established the theoretical framework and convergence guarantees, we now validate QA-Token empirically across two domains with distinct noise characteristics.

5. Empirical Validation

Setup: Results represent means over 10 trials with 95% CIs and Welch's t-test with Holm-Bonferroni correction (significance level $p_{\text{sig}} = 0.05$). Evaluation spans domain benchmarks, 7B-parameter foundation models, and ablation studies (complete details in Section G–Section G.3).

5.1. Genomics (QA-BPE-seq)

Data: 150bp paired-end reads (ART simulator (Huang et al., 2012), 30x coverage, doubled error rates), GRCh38 reference, GIAB HG002 truth set (Zook et al., 2016), CAMI II metagenome (Sczyrba et al., 2017). Details in Section G.

Baselines: We compare against (i) general-purpose tokenizers (BPE, SentencePiece (Kudo & Richardson, 2018), WordPiece), (ii) robustness-enhanced methods (BPE-dropout (Provilkov et al., 2020)), (iii) byte-level models (ByT5 (Xue et al., 2022), CANINE (Clark et al., 2021)), (iv) domain-standard k-mers (6-mer DNABERT (Ji et al., 2021)), and (v) neural approaches (CharFormer (Tay et al., 2022)).

Table 1. Downstream task performance for genomic tokenization. Values are means with 95% CI over $n = 10$ runs. Time: relative wall-clock (BPE=10.0×).

Method	Var. F1	Taxa F1	Recon.	Time
Standard BPE	.824±.004	.856±.005	.317±.010	10.0
SentencePiece	.837±.004	.872±.005	.301±.009	10.1
WordPiece	.829±.005	.863±.006	.308±.011	10.0
BPE-dropout	.841±.004	.878±.005	.295±.009	10.2
ByT5	.812±.006	.845±.007	.338±.012	25.3
CANINE	.818±.005	.852±.006	.325±.011	22.7
DNABERT-k	.851±.003	.889±.004	.287±.008	9.8
CharFormer	.856±.003	.893±.004	.279±.008	10.4
QA-BPE-seq	**.891±.004**	**.917±.003**	**.241±.007**	**10.2**

Table 2. Ablation Study for QA-BPE-seq (Variant F1). Values are means with 95% CI over $n = 10$ runs.*

Configuration	Var. F1	Δ(%)
QA-BPE-seq (Full)	**.891±.004**	—
w/o RL (Greedy w_{ab})	.862±.005	−3.3
w/o Quality ($R_Q = 0$)	.825±.004	−7.4
w/o Info. Reward ($R_I = 0$)	.872±.005	−2.1
w/o Adapt. Params	.857±.006	−3.8
w/o R_{bio}	.885±.004	−0.7
QualTok (Baseline)	.840±.005	−5.7

* "w/o RL (Greedy w_{ab})" uses full QA-Token merge score with learned α but selects merges greedily without RL policy optimization. "QualTok (Baseline)" additionally fixes adaptive parameters (α=0.5, uniform weights).

Quality Design: Phred scores with position decay, geometric mean aggregation, learned $\alpha = 0.72 \pm 0.03$, $\beta_{\text{pos}} = 0.014 \pm 0.002$.

Evaluation: (i) Variant calling via a Transformer model that takes token embeddings as features and predicts variant calls, evaluated against GIAB truth sets using `hap.py`; (ii) taxonomic classification (6-layer Transformer); (iii) sequence reconstruction (autoencoder), following established benchmarking protocols (Rumpf et al., 2023). Table 1 shows QA-BPE-seq outperforms all baselines ($p < 0.001$).

Key Insights: (i) QA-BPE-seq achieves 6.7 percentage point F1 improvement in variant calling (0.891 vs. 0.824 for BPE). (ii) Byte-level models fail catastrophically (2.5× slower, 7–9% lower accuracy). (iii) Emergent vocabulary aligns with biological units (codons, motifs) at high-quality regions without explicit supervision (vocabulary analysis in Section G).

Table 3. Ablation Study for QAT-QF (Return Pred. Acc. % and Sharpe Ratio). Means with 95% CI over $n = 10$ runs.*

Variant	Ret. Pred. (%)	Sharpe
Full Model	**68.3±0.5**	**1.72±0.07**
w/o Quality ($R_Q = 0$)	64.2±0.6	1.56±0.08
w/o Info. ($R_I = 0$)	65.1±0.5	1.61±0.07
w/o Pred. Power ($R_P = 0$)	63.9±0.6	1.49±0.09
w/o Complexity ($R_C = 0$)	66.8±0.4	1.73±0.06
Fixed α	65.4±0.5	1.65±0.07
Fixed γ	64.9±0.5	1.59±0.08
QualTok-QF (Baseline)	64.8±0.6	1.58±0.08

* "QualTok-QF (Baseline)" uses a simplified quality-aware merge score with fixed α=0.5 and uniform weights, without RL policy optimization or adaptive parameter learning.

5.2. Quantitative Finance (QAT-QF)

Dataset: We use high-frequency limit order book (LOB) data for the BTC/USD trading pair from LOBSTER (Huang & Polak, 2011), specifically reconstructed snapshots at 10 levels for the first quarter of 2023. The data is split chronologically into 70% for training, 15% for validation, and 15% for testing. Atomic elements are defined as sequences of 5 consecutive LOB events, encoded as tuples $(\Delta\text{mid}, \Delta\text{spread}, \text{vol\_imbalance}, \text{event\_type}, \Delta t)$ with discretization: price changes into 10 bins ($\pm$5 ticks), spread into 10 bins, volume imbalance into 5 signed bins, event types categorical (trade/cancel/limit order), time intervals into 5 log-spaced bins, yielding $|\Sigma| = 7{,}500$ atomic symbols (see Section D).

Baselines: QAT-QF is benchmarked against a diverse slate of tokenization and discretization methods relevant to financial time series.

- **General-Purpose:** Standard BPE, SentencePiece (Unigram LM mode), and BPE-dropout (Provilkov et al., 2020) to assess robustness.
- **Time-Series Specific:** Symbolic Aggregate approXimation (SAX) (Lin et al., 2003) (PAA=16, alphabet size=8) and Bag-of-SFA-Symbols (BOSS) (Schäfer, 2015), both widely used for symbolic time series representation.

The target vocabulary size for subword models is 16,000.

Evaluation: We assess (i) return prediction accuracy (5-minute mid-price return sign), (ii) volatility forecasting RMSE (5-minute realized volatility), (iii) market regime identification (2-state GARCH-HMM classification), and (iv) trading performance (Sharpe ratio (Sharpe, 1994) with 5bp transaction cost). Models use 2-layer LSTMs (128 hidden units) and PPO agents (Deng et al., 2016). See Section D and Section H.4 for implementation details.

Results: Table 4 presents results averaged over $n = 10$ runs. QAT-QF improves performance across all financial tasks ($p < 0.01$, Holm-Bonferroni corrected). The trading agent achieves Sharpe ratio of 1.72 ± 0.07 compared to 1.32 ± 0.05 for standard BPE (30% improvement). See ablation analysis in Table 3.

Table 4. Downstream task performance for financial tokenization. Values are means with 95% CI over $n = 10$ runs. Time: minutes per epoch.

Method	Ret. (%)	Vol.	Regime	Sharpe	Time
BPE	61.2±0.5	.014±.001	73.5±0.6	1.32±.05	15.0
SAX	58.9±0.6	.014±.001	75.2±0.5	1.29±.06	14.5
BOSS	62.3±0.4	.013±.001	78.4±0.4	1.45±.05	14.8
QAT-QF	**68.3±0.5**	**.010±.001**	**86.4±0.3**	**1.72±.07**	**15.2**

6. Foundation Model Validation

We validate QA-Token on domain benchmarks (Section 5) and now evaluate at foundation scale. We retrain state-of-the-art foundation models in genomics and finance to demonstrate that quality-aware tokenization improves how large models learn from noisy corpora, departing from traditional frequency-based approaches.

6.1. Metagenomics Foundation Model: METAGENE-1 7B

Setup: Re-tokenized METAGENE-1 (Liu et al., 2025) (7B parameters, 1.7T base pairs) with identical architecture/hyperparameters, comparing BPE vs QA-BPE-seq.

Quality-Aware Design: The tokenizer is trained on 2B base pairs (0.12% of corpus) using genomic quality metrics (Eq. 13, Section D) combining (i) Phred-based quality scores, (ii) conservation scores from k-mer analysis, (iii) GC-content deviation metrics, and (iv) secondary structure prediction confidence. The learned $\beta_{\text{pos}} = 0.014$ captures position-specific quality decay (see Section H.1 for implementation).

Training Budget: Both models process identical raw data volume (1.7T base pairs). The 15% token reduction means QA-BPE-seq completes epochs in fewer optimization steps while maintaining equal raw data exposure. Step-matched experiments (same optimization steps, where QA-BPE-seq processes 17.6% more raw data per step) show consistent improvements (Section G).

Pathogen Detection: QA-Token achieves state-of-the-art 94.53 MCC, surpassing the original METAGENE-1 by 1.57 points ($p < 0.001$). Consistent improvements across all five subtasks demonstrate robustness. Task-2 shows the largest gain (+2.04 MCC) on highly degraded metagenomic samples where quality awareness is most critical, validating

Table 5. Pathogen Detection benchmark (MCC). QA-Token achieves state-of-the-art.

Model	T-1	T-2	T-3	T-4	T-5	Avg
DNABERT	82.2	81.4	83.3	84.6	82.9	82.9
DNABERT-2	86.7	86.9	88.3	89.8	87.9	87.9
DNABERT-S	85.4	85.2	89.0	88.4	86.0	87.0
NT-2.5B-M	83.8	83.5	82.5	79.9	81.4	82.4
NT-2.5B-1k	77.5	80.4	79.8	78.4	79.0	79.0
HyenaDNA	78.7	79.1	80.4	81.2	79.9	79.9
METAGENE-1	92.1	90.9	93.7	95.1	94.0	93.0
+QA-Token	**93.8**	**93.0**	**95.1**	**96.2**	**94.5**	**94.5**
Δ	+1.7	+2.0	+1.4	+1.1	+0.6	+1.6

Table 6. Genome Understanding Evaluation (GUE): Multi-species benchmark.

Task	META-1	QA-Token	Δ	p
Regulatory Elements				
TF-Mouse (MCC)	71.4±0.8	**72.8±0.7**	+1.4	.002
TF-Human (MCC)	68.3±0.9	**69.9±0.8**	+1.6	.001
Promoter (MCC)	82.3±0.5	**85.5±0.4**	+3.2	<.001
Enhancer (AUC)	.876±.012	**.892±.010**	+.016	.003
Epigenetics				
H3K4me3 (MCC)	65.2±0.6	**66.8±0.5**	+1.6	.002
H3K27ac (MCC)	66.8±0.7	**68.2±0.6**	+1.4	.003
Methylation (AUC)	.823±.015	**.841±.013**	+.018	.004
Structure				
Splice Site (F1)	87.8±0.4	**89.5±0.3**	+1.7	<.001
RNA Structure	72.1±0.8	**73.9±0.7**	+1.8	.002
Variants				
COVID (F1)	72.5±0.6	**73.3±0.5**	+0.8	.018
SNP Effect	.684±.021	**.712±.018**	+.028	.001
Win Rate	46.4%	**57.1%**	**+10.7%**	—
Efficiency	370B	**315B**	**−15%**	—

our theoretical framework.

GUE Results: QA-Token improves performance across all categories (largest: +3.2 MCC promoter detection). 15% token reduction with performance gains indicates semantic coherence of quality-aware merging.

6.2. Financial Time-Series Foundation Model

Setup: 1.2B parameter model (24 layers, 2048 dim) inspired by TimesFM (Das et al., 2024) and Chronos (Ansari et al., 2024), using QAT-QF for noise handling.

Training Corpus: We train on 500 billion time-series observations spanning (i) high-frequency order book data (40%, 5 years millisecond-resolution across 50 liquid assets), (ii) daily OHLCV data (30%, 20 years for major indices), (iii) macroeconomic indicators (20%, 30 years G20 data), and (iv) alternative data (10%, sentiment scores, option flows, ETF compositions).

Table 7. Financial foundation model evaluation (100 test episodes).

Task	Zero-shot			Few-shot		
	BPE	QAT	Δ	BPE	QAT	Δ
Price Prediction						
Dir. 5m	52.3	**58.7**	+12	61.2	**68.3**	+12
Dir. 1h	51.8	**57.2**	+10	59.4	**65.8**	+11
Dir. 1d	50.9	**54.6**	+7	56.7	**61.2**	+8
Ret. MSE	1.00	**0.81**	−19	0.72	**0.60**	−18
Volatility						
Vol RMSE	.018	**.014**	−23	.013	**.010**	−27
GARCH Est.	.156	**.118**	−24	.098	**.071**	−28
Vol Regime	71.2	**79.8**	+12	82.3	**88.4**	+7
Microstructure						
Spread	.023	**.019**	−20	.018	**.013**	−25
Volume	31.2	**24.8**	−21	22.6	**17.3**	−24
Order Flow	.412	**.523**	+27	.567	**.681**	+20
Risk						
Regime F1	.673	**.751**	+12	.798	**.856**	+7
Drawdown	.682	**.743**	+9	.761	**.812**	+7
Tail Risk	.412	**.486**	+18	.523	**.598**	+14
Cross-Asset						
Corr. Pred.	.623	**.694**	+11	.712	**.768**	+8
Lead-Lag	58.3	**64.7**	+11	67.2	**73.1**	+9
Rotation	1.23	**1.41**	+15	1.52	**1.72**	+13
Avg. Δ	—	—	**+16%**	—	—	**+13%**

Quality-Aware Design: QAT-QF employs comprehensive market quality metrics (Eq. 14, Section D), combining liquidity, signal, stability, and information quality dimensions. The learned weights w_k adapt to different market regimes, with $\beta_{\text{vol}} = 0.50 \pm 0.05$ for volatility scaling (see Section H.2 for complete parameter settings).

Metrics: Dir. = directional accuracy (%); Ret. MSE = return prediction MSE (normalized to BPE=1.0); Vol RMSE = volatility forecast RMSE; Order Flow = order imbalance prediction R^2; Regime F1 = market regime classification F1; Tail Risk = VaR exceedance prediction F1; Rotation = sector rotation strategy Sharpe ratio.

Financial Results: QAT-QF achieves 7.3–27.0% zero-shot improvements, largest in volatility/microstructure tasks. Order flow imbalance (+27.0%) and regime detection (+11.6% F1) demonstrate QA-Token's noise-filtering capability, consistent with our information-theoretic optimality result (Theorem C.13). Implementation details in Section F–Section G.3.

Computational Costs: QA-Token requires 50–60 GPU-hours for vocabulary construction compared to minutes for standard BPE. However, this one-time cost is amortized across billions of inference operations: once constructed, the vocabulary imposes no additional inference overhead—tokenization speed is identical to BPE (∼10ms/sequence) as quality metrics are only used during construction. This efficiency is compatible with high-performance computing systems and in-storage processing architectures (Mansouri Ghiasi et al., 2022; Ghiasi et al., 2022; 2023; Mansouri Ghiasi et al., 2023; Ghiasi et al., 2024). For foundation models

where tokenization is performed once but affects billions of inference operations, the additional upfront cost is justified by substantial long-term gains; for small-scale applications or clean datasets, standard BPE may remain more practical.

7. Conclusion

QA-Token extends tokenization from frequency counting to quality-driven vocabulary construction, addressing limitations in processing noisy real-world data. We presented: (i) bilevel optimization with NP-hardness proof (Theorem 3.2, Section C.5), (ii) MDP formulation achieving $(1 - 1/e)$-approximation (Theorem E.4, Theorem E.5, Section E.7), (iii) Gumbel-Softmax enabling end-to-end learning (Theorem C.8, Section C.5). Our evaluation demonstrates consistent improvements: (1) genomics—6.7 pp F1 improvement, 94.53 MCC pathogen detection; (2) finance—30% Sharpe ratio increase; (3) foundation models achieve new benchmarks (analysis in Section G–Section G.3). As biological sequence archives scale to petabases (Karasikov et al., 2025) and variant prediction methods achieve unprecedented accuracy (Avsec et al., 2026), quality-aware tokenization becomes essential for bridging the gap between data availability and foundation model usability.

7.1. Scientific and Economic Impact

QA-Token enables utilization of massive noisy datasets previously considered unusable, fundamentally expanding the data frontier for foundation model training.

Scientific Acceleration in Genomics. The Sequence Read Archive (SRA) contains over 67 petabases of publicly available genomic data—equivalent to reading the human genome 22 million times—yet a substantial fraction remains underutilized due to quality heterogeneity (Leinonen et al., 2011). Recent infrastructure advances have made these petabase-scale archives full-text searchable at economical costs (Karasikov et al., 2025), and state-of-the-art methods like AlphaGenome now enable precise prediction of regulatory variant effects (Avsec et al., 2026). However, the gap between data *accessibility* and data *usability* for foundation model training persists: standard tokenization methods either discard low-quality reads entirely or propagate sequencing errors into learned representations. QA-Token bridges this gap by enabling quality-aware tokenization that can leverage the full breadth of available sequence data. We demonstrate three key applications: (1) *Pandemic surveillance*—environmental samples for pathogen monitoring contain 40–60% noise from contamination and sequencing errors; QA-Token directly trains on such noisy metagenomic data (Gollwitzer et al., 2023b;a; 2025a), achieving 94.53 MCC on pathogen detection and enabling real-time global pandemic monitoring using previously unusable environmental samples. (2) *Drug discovery*—long-read sequencing for structural variants has 10–15% error rates; our 6.7 percentage point F1 improvement in variant calling accelerates identification of drug targets from complex genomic rearrangements, complementing advances in regulatory variant prediction (Avsec et al., 2026). (3) *Evolutionary biology*—ancient DNA is heavily degraded with >50% damage; quality-aware tokenization preserves authentic ancient sequences while filtering damage, unlocking evolutionary insights from previously unanalyzable specimens.

Economic Impact in Finance. Global financial markets generate 5TB of data per day, with 40% containing microstructure noise from market fragmentation and latency; current approaches require expensive data cleaning infrastructure costing millions annually. QA-Token delivers quantifiable economic value: (1) *Algorithmic trading*—30% Sharpe ratio improvement translates to billions in additional returns for large funds; 27% better order flow prediction reduces execution costs by basis points worth millions daily. (2) *Risk management*—18% improvement in tail risk estimation could have prevented billions in losses during market crashes; 11.6% better regime detection enables faster portfolio rebalancing. (3) *Democratization*—smaller institutions can now compete without expensive data cleaning infrastructure, reducing barriers to entry for quantitative trading strategies.

Broader Societal Impact. Beyond genomics and finance, QA-Token has potential applications in: *Healthcare*—hospitals generate terabytes of noisy medical data daily; QA-Token enables training on real-world clinical data with artifacts, with potential to improve diagnostic accuracy and treatment recommendations, including applications in cancer treatment optimization (Gollwitzer et al., 2025b). *Climate science*—satellite imagery is often corrupted by cloud cover and atmospheric interference; QA-Token allows direct training on partially corrupted earth observation data, accelerating climate monitoring and prediction capabilities. *Infrastructure monitoring*—sensor networks produce petabytes of data with frequent failures; quality-aware tokenization enables robust anomaly detection despite sensor degradation, applicable to smart city applications and industrial IoT.

7.2. Limitations and Future Work

Limitations: (1) QA-Token requires domain-specific quality signals; domains without established metrics need custom design. (2) The vocabulary construction overhead limits rapid iteration during development. (3) Effective quality function design benefits from domain knowledge, though adaptive learning reduces sensitivity to initial choices.

Future Directions: (1) Universal quality metrics from data statistics (local entropy, consistency). (2) Online adaptation for streaming data. (3) Multimodal extension to vision-

language and audio-text. (4) Efficiency via distillation and pruning.

Impact Statement

Public sequence repositories now contain over 67 petabase pairs of raw sequencing data, with the European Nucleotide Archive doubling approximately every 45 months (Karasikov et al., 2025). Recent advances have made these petabase-scale archives full-text searchable at costs as low as $0.74 per queried megabase pair, demonstrating that the infrastructure for large-scale sequence analysis is maturing rapidly. However, a substantial fraction of this data remains underutilized for foundation model training due to quality heterogeneity. QA-Token bridges this gap between data *accessibility* and data *usability*, enabling quality-aware tokenization that can leverage the full breadth of available sequence data for foundation model training.

Genomics. We achieve 94.53 MCC on pathogen detection from environmental samples containing 40–60% noise, enabling real-time pandemic surveillance using previously unusable metagenomic data. Our 6.7 percentage point F1 improvement in variant calling accelerates drug target identification from complex genomic rearrangements with 10–15% sequencing error rates. The same technology could theoretically be misused for biosurveillance; we have designed QA-Token for research purposes with standard institutional safeguards.

Finance. Global financial markets generate 5TB of data per day, with 40% containing microstructure noise. Our 30% Sharpe ratio improvement translates to quantifiable returns for algorithmic trading, while 27% better order flow prediction reduces execution costs. Enhanced trading performance raises concerns about market fairness; QA-Token provides incremental improvements within existing regulatory frameworks.

Resources. The 50–60 GPU-hour vocabulary construction cost is substantially lower than foundation model training costs, making QA-Token accessible to researchers with modest computational budgets. The highly compressed quality-aware vocabularies are portable for further analysis.

Reproducibility Statement

We provide comprehensive details throughout the paper and appendices.

Theoretical contributions: All theorems and propositions include complete proofs (Section C.5, Section C.2, Section C.5, Section C.5, Section C.9) with explicit assumptions (Section C.6) and convergence guarantees (Section E.4, Section E).

Algorithms: Complete pseudocode for RL policy optimization (Algorithm 1), adaptive parameter learning (Algorithm 2), and final vocabulary construction (Algorithm 3) are provided in Section E.

Implementation: Domain-specific quality metrics with exact formulas (Section 2, Section D), hyperparameters for all models (Section H.1, Section H.2), and computational requirements (Section G.3) are fully specified.

Experimental protocol: Statistical methodology including 10 independent trials, 95% confidence intervals, Welch's t-test with Holm-Bonferroni correction, and effect sizes are detailed in Section 5 and Section G. Dataset specifications, preprocessing steps, and evaluation metrics are provided in Section G–Section I.2.

Baselines: Nine baseline methods with implementation details and hyperparameters are described in Section 5 and Section I.4.

Code release: We will provide a GitHub repository with all source code, trained models, and scripts to reproduce results.

Conflict of Interest Statement

A.E.G. and D.d.G. are co-founders and shareholders of Anto Biosciences (YC F25).

D.A.S., P.L., and A.N.d.l.C. declare no competing interests.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, May 2017.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. The distribution of realized exchange rate volatility. *Journal of the American statistical association*, 96(453): 42–55, 2001.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Avsec, Ž., Latysheva, N., Cheng, J., et al. Advancing regulatory variant effect prediction with AlphaGenome. *Nature*, 649:1206–1218, jan 2026. doi: 10.1038/s41586-025-10014-0.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. Noisy text analytics. In *Proceedings of the Australasian Language Technology Association Workshop 2013*, pp. 1–10, 2013.

Barbieri, F., Camacho-Collados, J., Ronzano, F., Espinosa-Anke, L., Ballesteros, M., Basile, V., Patti, V., and Saggion, H. Semeval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 24–33, 2018.

Barbieri, F., Camacho-Collados, J., Espinosa-Anke, L., and Neves, L. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*, 2020.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2007.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. Neural combinatorial optimization with reinforcement learning. In *International Conference on Learning Representations*, 2016.

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *Operations Research*, 69(6): 1744–1767, Dec 2021. doi: 10.1287/opre.2021.0014.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. In *Transactions of the Association for Computational Linguistics*, volume 5, pp. 135–146, 2017.

Bolte, J., Le, Q.-T., Pauwels, E., and Vaiter, S. Geometric and computational hardness of bilevel programming. *Mathematical programming*, 2024. doi: 10.1007/s10107-025-02229-w.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency, 2009.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Cen, S. and Chi, Y. Global convergence of policy gradient methods in reinforcement learning, games and control. *arXiv preprint arXiv:2310.05230*, 2023.

Chai, B. Y., Wang, Z., and Sachan, M. The curse of tokenization. *arXiv preprint arXiv:2402.07831*, 2024.

Church, K. W. and Hanks, P. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

Clark, J. H., Garcia, D., Botha, J., Lee, K., Luong, M.-T., and Le, Q. V. Canine: Pre-training an efficient tokenization-free encoder for language representation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2647–2661, 2021.

Das, A., Kong, W., Leach, A., Sen, R., and Yu, R. Timesfm: A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2024.

Dempe, S. *Bilevel Optimization: Theory, Algorithms and Applications*, volume 161 of *Springer Optimization and Its Applications*. Springer, Berlin, Germany, 2020. doi: 10.1007/978-3-030-33566-3.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Ewing, B., Hillier, L., Wendl, M. C., and Green, P. Basecalling of automated sequencer traces using phred. i. accuracy assessment. *Genome research*, 8(3):175–185, 1998.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Gençay, R., Selçuk, F., and Whitcher, B. *An introduction to wavelets and other filtering methods in finance and economics*. Elsevier, San Diego, 2001.

Ghiasi, N. M., Park, J., Mustafa, H., Kim, J., Olgun, A., Gollwitzer, A., Cali, D. S., Firtina, C., Mao, H., Alserr, N. A., et al. Genstore: In-storage filtering of genomic data for high-performance and energy-efficient genome analysis. In *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 283–287. IEEE, 2022.

Ghiasi, N. M., Sadrosadati, M., Mustafa, H., Gollwitzer, A., Firtina, C., Eudine, J., Ma, H., Lindegger, J., Cavlak, M. B., Alser, M., et al. Metastore: High-performance metagenomic analysis via in-storage computing. *arXiv preprint arXiv:2311.12527*, 2023.

Ghiasi, N. M., Sadrosadati, M., Mustafa, H., Gollwitzer, A., Firtina, C., Eudine, J., Mao, H., Lindegger, J., Cavlak, M. B., Alser, M., et al. Megis: High-performance, energy-efficient, and low-cost metagenomic analysis with in-storage processing. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 660–677. IEEE, 2024.

Gollwitzer, A., Alser, M., Bergtholdt, J., Lindegger, J., Rumpf, M.-D., Firtina, C., Mangul, S., and Mutlu, O. Metafast: Enabling fast metagenomic classification via seed counting and edit distance approximation. *arXiv*, pp. 2311–02029, 2023a.

Gollwitzer, A. E., Alser, M., Bergtholdt, J., Lindegger, J., Rumpf, M.-D., Firtina, C., Mangul, S., and Mutlu, O. Metatrinity: Enabling fast metagenomic classification via seed counting and edit distance approximation. *arXiv preprint arXiv:2311.02029*, 2023b.

Gollwitzer, A. E., Subramanian, D. A., Tucker, I., and Traverso, G. Metaomics-10t: The foundational dataset to unlock causal modeling of microbial ecosystems. In *NeurIPS 2025 AI for Science Workshop*, 2025a.

Gollwitzer, A. E., Subramanian, D. A., Tucker, I., and Traverso, G. Steering the evolutionary game: Hierarchical control of therapeutic resistance in cancer treatment. In *NeurIPS 2025 AI for Science Workshop*, 2025b.

Golovin, D. and Krause, A. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2675–2683, 2011.

Grne, C. and Wulf, L. Completeness in the polynomial hierarchy for many natural problems in bilevel and robust optimization. *Conference on Integer Programming and Combinatorial Optimization*, 2023. doi: 10.1007/978-3-031-93112-3_19.

Hamilton, J. D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pp. 357–384, 1989.

Han, B., Cook, P., and Baldwin, T. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 368–378, 2013.

Hansen, P. R. and Lunde, A. Realized variance and market microstructure noise. *Journal of Business & Economic Statistics*, 24(2):127–161, 2006.

Harrow, J., Frankish, A., Gonzalez, J. M., Tapanari, E., Aken, B., Barrell, D., Mudge, J. M., FRecognision, E., GCoil, A., LNCipedia, A., et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9):1760–1774, 2012.

Hasbrouck, J. Measuring the information content of stock trades. *The Journal of Finance*, 46(1):179–207, 1991.

Heafield, K. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197. Association for Computational Linguistics, 2011.

Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Neettiyath, U., and Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):1–17, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015.

Huang, R. and Polak, T. Lobster: Limit order book reconstruction system. *Available at SSRN 1920143*, 2011.

Huang, W., Li, L., Myers, J. R., and Marth, G. T. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28 (4):593–594, 2012.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

J.P. Morgan. Riskmetrics technical document. Technical report, J.P. Morgan/Reuters, 1996.

Karasikov, M., Mustafa, H., Danciu, D., Bosshard, L., Zimmermann, M., Schütze, K., Kahles, A., and Rätsch, G. Efficient and accurate search in petabase-scale sequence repositories. *Nature*, 647:1036–1044, 2025. doi: 10.1038/s41586-025-09603-w.

Karp, R. M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103. Springer, 1972.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, 2018.

Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.

Leinonen, R., Sugawara, H., Shumway, M., and International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic Acids Research*, 39 (suppl_1):D19–D21, 2011.

Li, J., Park, Y.-B., Song, Y.-S., and Park, S.-K. An empirical study of tokenization strategies for various korean nlp tasks. In *Proceedings of the 12th language resources and evaluation conference*, pp. 6813–6819, 2020.

Libovick'y, J. and Sachan, M. Semantic segmentation for improving the performance of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4930–4945, 2024.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520. Association for Computational Linguistics, 2011.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. Symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11, 2003.

Liu, O. et al. METAGENE-1: Metagenomic foundation model for pandemic monitoring. *arXiv preprint arXiv:2501.02045*, jan 2025.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Madhavan, A. Market microstructure: A survey. *Journal of financial markets*, 3(3):205–258, 2000.

Mansouri Ghiasi, N., Park, J., Mustafa, H., Kim, J., Olgun, A., Gollwitzer, A., Senol Cali, D., Firtina, C., Mao, H., Almadhoun Alserr, N., et al. Genstore: A high-performance in-storage processing system for genome sequence analysis. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 635–654, 2022.

Mansouri Ghiasi, N., Sadrosadati, M., Mustafa, H., Gollwitzer, A., Firtina, C., Eudine, J., Ma, H., Lindegger, J., Banu Cavlak, M., Alser, M., et al. Metastore: High-performance metagenomic analysis via in-storage computing. *arXiv e-prints*, pp. arXiv–2311, 2023.

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 31–41, 2016.

Mohammad, S., Bravo-Marquez, F., Salameh, M., and Kiritchenko, S. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pp. 1–17, 2018.

Moody, J. and Saffell, M. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, 20(1):1–18, 2001.

Moody, J. and Wu, L. Learning to trade via direct reinforcement. In *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1741–1746. IEEE, 1998.

Nguyen, D. Q., Vu, T., and Nguyen, A. T. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 9–14, 2020.

Owen, A. B. *Monte Carlo theory, methods and examples*. Stanford University, 2013. Available at https://artowen.su.domains/mc/.

Provilkov, I., Emelyanenko, D., and Voita, E. Bpe-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1882–1892, 2020.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2015.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Rosenthal, S., Farra, N., and Nakov, P. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 502–518, 2017.

Rumpf, M.-D., Alser, M., Gollwitzer, A. E., Lindegger, J., Almadhoun, N., Firtina, C., Mangul, S., and Mutlu, O. Sequencelab: A comprehensive benchmark of computational methods for comparing genomic sequences. *arXiv preprint arXiv:2310.16908*, 2023.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. Policy distillation, 2016.

Schäfer, P. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dr"oge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., et al. Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods*, 14(11):1063–1071, 2017.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sharpe, W. F. The sharpe ratio. *Journal of portfolio management*, 21(1):49–58, 1994.

Sherry, S. T., Ward, M.-H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., and Sirotkin, K. dbsnp: the ncbi database of genetic variation. *Nucleic acids research*, 29 (1):308–311, 2001.

Subramanyam, A., Chen, Y., and Grossman, R. L. Scaling laws revisited: Modeling the role of data quality in language model pretraining. *arXiv.org*, 2025. doi: 10.48550/arXiv.2510.03313.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tay, Y., Tran, V. Q., Ruder, S., Gupta, J., Liu, L., Chung, J., Turner, S., Wang, Z., Williams, D., Casas, D. G., et al. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2022.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

Van Hee, C., Lefever, E., and Hoste, V. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 39–50, 2018.

Wenger, A. M., Peluso, P., Rowell, W. J., Chang, P.-C., Hall, R. J., Concepcion, G. T., Ebler, J., Fungtammasan, A., Kolesnikov, A., Olson, N. D., et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature biotechnology*, 37(10):1155–1162, 2019.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

Yue, Y. et al. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, apr 2025. Presented at NeurIPS 2025 (Oral), ICML 2025 AI4Math Workshop Best Paper.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 75–86, 2019.

Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., Weng, Z., and Salit, M. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, 3(1):1–19, 2016.

Supplementary Information

A. Notation

To ensure clarity and rigor, we define our mathematical notation in Table 8. We distinguish between atomic (indivisible) elements and tokens (sequences of atomic elements or other tokens).

Table 8. Table of Notation

Symbol	Definition
Σ	Base alphabet of atomic elements (e.g., characters, DNA bases).
s_i	An atomic element from Σ.
q_i	Scalar quality score of an atomic element s_i, where $q_i \in [0, 1]$.
t, a, b	Tokens, which are sequences of atomic elements.
V_k	Vocabulary at merge step k.
$f(t)$	Frequency of token t in the corpus.
$\|t\|$	Length of token t in atomic elements.
$n_\sigma(t)$	Count of atomic element $\sigma \in \Sigma$ within token t.
$H(t)$	Empirical entropy of token t: $H(t) = -\sum_\sigma \frac{n_\sigma(t)}{\|t\|} \log \frac{n_\sigma(t)}{\|t\|}$.
$\boldsymbol{q}_t$	Vector of quality scores for token t (in multi-dimensional domains).
q_t	Aggregated scalar quality score of token t, derived from its constituents.
$\bar{q}_{ab}$	Average quality of constituent tokens a, b, defined as $(q_a + q_b)/2$.
α	Learnable exponent controlling sensitivity to quality in the merge score.
w_{ab}	Quality-aware merge score for the token pair (a, b).
θ_{adapt}	Vector of all learnable adaptive parameters in the framework.
π_{θ_π}	Reinforcement learning policy for selecting merges, parameterized by θ_π.
L_{task}	Loss function of the downstream machine learning task.
$\mathcal{J}(\mathcal{T})$	Global objective function for the tokenization process (Eq. 1).

B. Related Work

QA-Token intersects with, and extends upon, research in subword tokenization, noisy data handling, reinforcement learning for sequential optimization, and adaptive or differentiable modeling techniques. Table 9 provides a comparative overview, situating QA-Token relative to existing approaches and highlighting its unique synthesis of explicit quality integration, RL-based optimization of merges, and adaptive learning of the tokenization process parameters. The key distinction of QA-Token's adaptive parameter learning is its focus on optimizing parameters governing the tokenization *process* itself (like quality sensitivity or reward component weights), rather than solely adapting the vocabulary content or segmentation boundaries within a fixed merge logic.

Table 9. Comparison of QA-Token with Representative Tokenization Approaches.

Method	Explicit Quality Integration	Optimization Method	Adaptive Params (Learned Process?)	Downstream Aware (via Reward/Loss)	Domain Noise Model (Explicit?)	Vocabulary Type
Standard BPE/WP/SP (Sennrich et al., 2016; Wu et al., 2016; Kudo & Richardson, 2018)	No	Frequency	No	No	No	Subword
BPE-Dropout (Provilkov et al., 2020)	No	Freq.+Stochastic	No	No	No	Subword
Char/Byte Models (Xue et al., 2022; Clark et al., 2021)	No	N/A (Fixed)	No	Yes (via model)	Implicit	Char/Byte
Gradient-based (Tay et al., 2022)	No	Gradient	Yes (Segmenter)	Yes	Implicit	Char/Subword
Semantic Tokenizers (Libovický & Sachan, 2024)	No	Semantics+Freq	No	Indirectly	No	Subword
QA-Token (Ours)	**Yes**	**RL (Policy) + Gradient (HPs)**	**Yes (Process HPs: $\alpha, \lambda_i, w_j, \beta_k$)**	**Yes (via Reward for RL, $L_{\text{downstream}}$ for HPs)**	**Yes (via Q, R)**	**Subword**

*Note: "Adaptive Params (Learned Process?)" refers to learning parameters governing the tokenization *process* itself (like QA-Token's $\alpha, \beta_k, \lambda_i, w_j$), not just the vocabulary content or segmentation boundaries. QA-Token uses RL to optimize the merge policy and gradient-based methods to optimize these process hyperparameters.*

Subword Tokenization Algorithms: The prevailing paradigm relies on frequency-based greedy merging procedures, exemplified by BPE (Sennrich et al., 2016), WordPiece (Wu et al., 2016) (which optimizes data likelihood), and SentencePiece (Kudo & Richardson, 2018) (which operates directly on raw text). While computationally efficient and broadly effective, their fundamental mechanism ignores sequence quality, providing the primary motivation for our work. BPE-dropout

(Provilkov et al., 2020) introduces stochasticity during the merge process as a form of regularization to enhance robustness, but it does not use explicit quality signals. Unigram language models (Kudo, 2018) present a probabilistic alternative, yet they still primarily depend on frequency and likelihood objectives without explicit quality awareness.

Handling Noisy and Domain-Specific Data: Considerable research focuses on modeling noise within particular application domains. In genomics, Phred scores (Ewing et al., 1998) are standard quality indicators, and specialized models aim to account for sequencing errors (Heinzinger et al., 2019). In NLP, extensive work on social media text addresses lexical variation, misspellings, and slang through techniques like text normalization (Han et al., 2013; Li et al., 2020) and explicit noise modeling (Baldwin et al., 2013). Financial time series analysis frequently employs filtering methods (Gençay et al., 2001), microstructure modeling (Madhavan, 2000; Hasbrouck, 1991), and regime-switching models (Hamilton, 1989) to manage inherent noise and non-stationarity. QA-Token distinguishes itself by offering a *unified tokenization framework* that directly integrates such domain-specific quality and noise considerations into the token construction process itself, rather than addressing noise solely as a separate downstream modeling challenge. The notion of the "curse of tokenization" (Chai et al., 2024), which highlights the downstream impact of tokenization choices on LLM robustness, further underscores the need for quality-aware approaches.

Reinforcement Learning for Sequential Optimization: RL offers a robust framework for sequential decision-making under uncertainty (Sutton & Barto, 2018). It finds successful application in various optimization problems involving sequences, including text generation (Ranzato et al., 2015), combinatorial optimization (Bello et al., 2016), and financial strategy optimization (Moody & Wu, 1998; Moody & Saffell, 2001). We uniquely formulate the tokenization vocabulary construction process as an RL problem where merge operations constitute actions selected by a learned policy to maximize a cumulative reward signal reflecting token quality, information content, complexity, and estimated utility. This formulation allows for optimizing complex, potentially non-differentiable objectives related to the quality of the final tokenization outcome. The rewards themselves are shaped by adaptively learned parameters (Section 4.3).

Adaptive and Differentiable Tokenization: Acknowledging the limitations inherent in static tokenizers, researchers explore adaptive and learnable alternatives. Gradient-based approaches (Tay et al., 2022) learn segmentation parameters end-to-end concurrently with downstream tasks, often operating at the character level. Semantic tokenization (Libovick'y & Sachan, 2024) uses word meanings to inform the segmentation process. QA-Token integrates adaptive learning distinctively: it learns hyperparameters ($\alpha, \beta_k, w_j, \lambda_i, \dots$) that directly govern the quality-aware merge decisions and the RL agent's reward structure. This learning is enabled by Gumbel-Softmax relaxation (Jang et al., 2017; Maddison et al., 2017) for making merge choices differentiable with respect to these hyperparameters when optimizing a downstream task loss (via composite logits defined in Equation 37). This enables the fundamental *tokenization logic* to adapt based on observed data properties and task feedback, co-evolving with the RL agent's policy. Meta-learning (Finn et al., 2017) provides a potential mechanism, explored conceptually within QA-Token (see Appendix E.5), to further accelerate adaptation across heterogeneous data sources (e.g., different social media platforms).

In essence, QA-Token synthesizes concepts from these related areas but provides a unique combination: explicit quality integration within the merge decision, optimization of the merge sequence via RL using a multi-faceted reward signal, and adaptive learning of core process parameters that define this reward and merge logic, demonstrating applicability across diverse, noisy domains.

C. Theoretical Framework and Proofs

C.1. Quality Metric Proofs

Proposition C.1 (Boundedness and Continuity of Quality Functions)**.** *All domain-specific quality functions $q_t \in [0, 1]$ are:*

1. *Bounded: $0 \leq q_t \leq 1$ for all tokens t*
2. *Continuous: Lipschitz continuous in their arguments*
3. *Monotonic: Quality decreases with increasing noise/error*

Proof. **Boundedness:** For genomics, the geometric mean of values in $[0, 1]$ remains in $[0, 1]$. For finance, the convex combination of bounded components $q_{k,t} \in [0, 1]$ with $\sum_k w_k = 1$ yields $q_t^{\text{finance}} \in [0, 1]$.

Lipschitz continuity: For genomics (geometric mean on $[\epsilon_Q, 1]^n$), the chain rule via logarithmic transformation yields Lipschitz constant $L_g = 1/(\sqrt{n} \cdot \epsilon_Q)$. For finance, the weighted sum of Lipschitz component functions has $L_f \leq \max_k L_k$.

Monotonicity: For any noise injection η with $\eta(q) \leq q$, both aggregations (geometric and arithmetic means) preserve the ordering: noisier inputs yield lower quality scores. □

C.2. Merge Score Derivation

Lemma C.2 (First-Order Approximation)**.** *The marginal gain in objective $\mathcal{J}$ from merge $(a, b) \mapsto ab$ admits the decomposition:*

$$\boxed{\Delta\mathcal{J}(a, b) = \lambda_{LM}\Delta\mathcal{L}_{LM} - \lambda_{comp}\Delta\Phi + \lambda_{qual}\Delta Q + O(\epsilon^2)} \tag{4}$$

where $\epsilon = 1/|\mathcal{S}|$ represents the corpus-normalized perturbation.

Proof. The marginal gain decomposes into three components following standard vocabulary optimization analysis (Sennrich et al., 2016).

Language Model Component: The change $\Delta\mathcal{L}_{\text{LM}} \approx f(a, b) \cdot \text{PMI}(a, b)$ follows from the pseudo-likelihood approximation, where PMI (Pointwise Mutual Information) captures statistical association (Church & Hanks, 1990).

Complexity Component: By MDL principles (Rissanen, 1978), merging reduces vocabulary complexity: $\Delta\Phi = -\log|V| - 1 + O(|V|^{-1})$. This compression benefit is absorbed into the PMI term, which also favors frequent co-occurrences.

Quality Component: For concave aggregator $g(x) = (x+\epsilon_Q)^\alpha$, Jensen's inequality yields $g(\bar{q}_{ab}) \geq \frac{1}{2}(g(q_a)+g(q_b))$. The dominant quality contribution is $\Delta Q_+ = f(a, b) \cdot g(\bar{q}_{ab})$ where $\bar{q}_{ab} = (q_a + q_b)/2$. Normalization errors are $O(f(a, b)/T)$, negligible for typical corpora. □

C.3. Derivation of the Optimal Merge Score

Theorem C.3 (Quality-Aware Merge Score — Principled Heuristic)**.** Motivated by *the first-order approximation of $\Delta\mathcal{J}$ (Lemma C.2), we propose the following quality-aware merge score as a* ***principled heuristic****:*

$$\boxed{w_{ab} = \frac{f(a,b)}{f(a)f(b)+\epsilon_f} \cdot (\bar{q}_{ab} + \epsilon_Q)^\alpha \cdot \psi(a, b)} \tag{5}$$

where:

- $f(\cdot)$ *denotes frequency in the corpus*
- $\bar{q}_{ab} = (q_a + q_b)/2$ *is the average constituent quality*
- $\alpha \in (0, 1]$ *is a learnable parameter controlling quality sensitivity*
- $\epsilon_f, \epsilon_Q > 0$ *ensure numerical stability*
- $\psi(a, b) \in [0, 1]$ *encodes domain-specific constraints*

Note: *The derivation below involves two principled approximations (Steps 4–5) that trade mathematical exactness for computational tractability. The resulting score preserves key monotonicity properties and is calibrated end-to-end via downstream task performance.*

Proof. From Lemma C.2, the marginal gain is $\Delta\mathcal{J}(a, b) = \lambda_{\text{LM}} f(a, b) \cdot \text{PMI}(a, b) + \lambda_{\text{qual}} f(a, b) g(\bar{q}_{ab}) + O(1/|V|)$, where the complexity term is absorbed into PMI (both favor frequent co-occurrences).

Per-occurrence normalization: Following the design principle of BPE (Sennrich et al., 2016), we normalize by frequency to capture per-occurrence information gain. Applying the exponential transform (monotonic, preserves rankings): $\exp(\Delta\mathcal{J}/f(a, b)) \propto \frac{f(a,b)}{f(a)f(b)+\epsilon_f} \cdot \exp(\frac{\lambda_{\text{qual}}}{\lambda_{\text{LM}}} g(\bar{q}_{ab}))$

Power-law approximation: We replace $\exp(\lambda \cdot g(q))$ with $(\bar{q}_{ab} + \epsilon_Q)^{\tilde{\alpha}}$ where $\tilde{\alpha}$ is learned end-to-end. This preserves monotonicity in $\bar{q}_{ab}$ and subsumes the unknown ratio $\lambda_{\text{qual}}/\lambda_{\text{LM}}$. The final score is: $w_{ab} = \frac{f(a,b)}{f(a)f(b)+\epsilon_f} \cdot (\bar{q}_{ab} + \epsilon_Q)^{\alpha} \cdot \psi(a,b)$

Monotonicity guarantees: $\partial w_{ab}/\partial \bar{q}_{ab} > 0$ and $\partial w_{ab}/\partial \text{PMI} > 0$, ensuring quality-increasing and statistically-associated merges are preferred. End-to-end learning of α calibrates the heuristic. □

C.4. Key Insights from the Derivation

1. **PMI Foundation:** The frequency term $\frac{f(a,b)}{f(a)f(b)+\epsilon_f}$ approximates Pointwise Mutual Information, capturing statistical association.
2. **Quality Modulation:** The quality term $(\bar{q}_{ab} + \epsilon_Q)^{\alpha}$ multiplicatively adjusts the PMI-based score, up-weighting high-quality merges.
3. **Learnable Sensitivity:** The parameter α controls the relative importance of quality vs. frequency:
   - $\alpha = 0$: Reduces to standard PMI-based tokenization
   - $\alpha > 0$: Increasing weight on quality signals
   - Learned via gradient descent to optimize downstream performance
4. **Domain Flexibility:** The factor $\psi(a,b)$ allows incorporation of domain knowledge without modifying the core framework.

This derivation shows that the quality-aware merge score is a *principled heuristic* motivated by first-principles analysis of the bilevel objective, rather than an ad-hoc combination of frequency and quality terms.

C.5. Theory Proofs

Proof of Theorem 3.2 (Computational Complexity). The bilevel optimization problem is NP-hard by polynomial-time reduction from Weighted Set Cover (Karp, 1972). The reduction maps sets to corpus sequences and set cover cost to vocabulary complexity: given a WSC instance $(U, \mathcal{S}, \{c_i\})$, construct alphabet $\Sigma = U \cup \{\$\}$, corpus sequences σ_i for each set S_i, and uniform quality scores. With $\lambda_{\text{qual}} = 0$, optimal tokenizations correspond bijectively to optimal set covers.

For stronger complexity results establishing Σ_2^p-hardness of general bilevel programs, see (Bolte et al., 2024; Grne & Wulf, 2023; Dempe, 2020). The worst-case exhaustive search complexity is $O(|\Sigma|^K \cdot K! \cdot N \cdot n \cdot |\Theta|)$, accounting for the space of merge sequences, merge orderings, and downstream model optimization.

□

Proposition C.4 (Boundedness and Lipschitzness of w_{ab})**.** *Under assumptions (A1)-(A2), the quality-aware merge score w_{ab} is bounded and Lipschitz continuous in (q_a, q_b).*

Proof. **Boundedness:** By (A1), $f(a,b)/(f(a)f(b) + \epsilon_f) \leq C_f/\epsilon_f$. With $\bar{q}_{ab} \in [0,1]$ and $\psi \in [0,1]$, we have $w_{ab} \leq C_f(1+\epsilon_Q)^{\alpha}/\epsilon_f =: C_w$.

Lipschitz continuity: By chain rule on compositions of bounded functions on compact domains, w_{ab} is L_w-Lipschitz in (q_a, q_b) with $L_w = C_f L_g/\epsilon_f$. For $\alpha = 1$, $L_g = 1/\sqrt{2}$; for $0 < \alpha < 1$, $L_g \leq \alpha \epsilon_Q^{\alpha-1}/\sqrt{2}$. The regularization ϵ_Q ensures numerical stability. □

Proposition C.5 (Stability of EMA Normalization)**.** *Under assumptions (A1) and $\epsilon_R > 0$, the EMA-based normalization maintains $\sigma_{j,t}^{run} > 0$ almost surely for non-degenerate reward streams.*

Proof. The result follows from standard EMA convergence theory (Robbins-Monro). Under (A1), raw rewards have non-degenerate distribution $\text{Var}(X_t) > 0$. The EMA variance update preserves positivity: if $\text{Var}_{j,t-1}^{\text{run}} > 0$, then $\text{Var}_{j,t}^{\text{run}} \geq (1 - \beta_{\text{norm}})\text{Var}_{j,t-1}^{\text{run}} > 0$.

With $\sum_t \beta_{\text{norm},t} = \infty$ and $\sum_t \beta_{\text{norm},t}^2 < \infty$, the running variance converges a.s. to $\text{Var}(X) > 0$, ensuring $\sigma_{j,t}^{\text{run}} > 0$. □

Proposition C.6 (Convergence of PPO Objective)**.** *Under assumptions (A1)-(A4), PPO converges to a stationary point of $J(\pi; \theta_{adapt}^{(0)})$.*

Proof. Under (A1)–(A4), the standard PPO conditions hold (Schulman et al., 2017): bounded rewards ($|R(s,a)| \leq R_{\max}$), compact state space, finite action space, and differentiable policy. The clipped surrogate objective ensures bounded gradients $\|\nabla_\theta L^{\text{CLIP}}\|_2 \leq G_{\max}$.

With learning rate $\eta_t = \eta_0/\sqrt{t}$ satisfying $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, global convergence to stationary points at rate $O(1/\sqrt{T})$ follows from (Bhandari & Russo, 2021; Cen & Chi, 2023). □

Proposition C.7 (Consistency and Boundedness of Stage 2 Gradients)**.** *Under assumptions (A1)–(A3), the Gumbel-Softmax estimator yields consistent gradients with bounded variance.*

Proof. The Gumbel-Softmax gradient properties follow from (Jang et al., 2017; Maddison et al., 2017). Under (A1)–(A3), the composite logits ℓ_{ab} are bounded by $L_{\max} = C_w + \sum_j |\lambda_j| R_{\max}$. The Gumbel-Softmax Jacobian satisfies $\|\partial y_i/\partial \ell_j\| \leq 1/\tau$, yielding bounded gradients $\|\nabla_{\theta_{\text{adapt}}} L_{\text{task}}\| \leq L_{\max}/\tau \cdot \|\nabla_y L_{\text{task}}\|$.

As $\tau \to 0$, the estimator converges to REINFORCE (Williams, 1992). The bias-variance tradeoff is: $\text{Bias}(\tau) = O(\tau^2)$, $\text{Var}(\tau) = O(1/\tau^2)$. The optimal temperature $\tau_{\text{opt}} \propto T^{-1/4}$ for T samples balances these terms. □

Theorem C.8 (Gumbel-Softmax Properties)**.** *Let $\pi = (\pi_1, \ldots, \pi_k)$ be a categorical distribution with k categories. The Gumbel-Softmax distribution with temperature $\tau > 0$ satisfies:*

1. ***Consistency:*** *As $\tau \to 0$, the samples converge to one-hot vectors from Categorical(π)*
2. ***Differentiability:*** *The reparameterization provides continuous gradients with respect to π*
3. ***Bias-Variance Tradeoff:*** *Bias $O(\tau^2)$, Variance $O(1/\tau^2)$*

Proof. All three properties are established in (Jang et al., 2017; Maddison et al., 2017). We summarize the key arguments.

Property 1 (Consistency): By the Gumbel-Max trick, $\arg\max_i(\ell_i + g_i) \sim \text{Categorical}(\text{softmax}(\boldsymbol{\ell}))$ for $g_i \sim \text{Gumbel}(0,1)$. As $\tau \to 0$, the Gumbel-Softmax samples $y_i = \exp((\ell_i + g_i)/\tau)/\sum_j \exp((\ell_j + g_j)/\tau)$ concentrate on one-hot vectors almost surely by the continuous mapping theorem.

Property 2 (Differentiability): For $\tau > 0$, y_i is C^∞ in ℓ_j, enabling reparameterized gradients. The expectation $\mathbb{E}_g[y_i] = \text{softmax}(\boldsymbol{\ell}/\tau)_i$ introduces bias that vanishes as $\tau \to 0$. The annealing schedule $\tau_t \to 0$ ensures asymptotic consistency.

Property 3 (Gradient Bounds): The Jacobian satisfies $\partial y_i/\partial \ell_j = (1/\tau) y_i(\delta_{ij} - y_j)$, yielding $\|\nabla_{\boldsymbol{\ell}} \mathbf{y}\|_F \leq 1/\tau$. □

C.6. Assumptions

We formalize the assumptions used throughout the theoretical analysis:

Assumption A1 (Bounded Frequencies): There exists $C_f > 0$ such that for all tokens a, b:

$$0 \leq f(a), f(b), f(a,b) \leq C_f$$

Assumption A2 (Bounded Qualities): All quality scores satisfy $q \in [0,1]$, and the quality aggregation function is L_Q-Lipschitz continuous.

Assumption A3 (Bounded Rewards): Raw reward components are bounded: $|R_j^{\text{raw}}| \leq R_{\max}$ for all j.

Assumption A4 (Regular Learning Rates): The learning rate schedules satisfy: - PPO: $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$ - Adaptive learning: $\eta_t = O(1/\sqrt{t})$

C.7. Theory Extensions

Definition C.9 (Assumptions for Approximation Guarantee)**.** The $(1 - 1/e)$ approximation guarantee requires the following structural assumptions:

(A1) **Adaptive Monotonicity:** For any partial realization ψ and merge (a,b): $\Delta_F((a,b)|\psi) \geq 0$, where Δ_F denotes the marginal gain.

(A2) **Adaptive Submodularity:** For realizations $\psi \preceq \psi'$ (where $\preceq$ denotes extension): $\Delta_F((a,b)|\psi) \geq \Delta_F((a,b)|\psi')$ (diminishing returns).
(A3) **Constraint independence:** $\psi(a,b)$ is history-independent.
(A4) **Candidate pool regularity:** $\mathbb{P}[(a,b) \in PQ_t] \geq \delta > 0$ for all valid pairs.
(A5) **Quality stability:** $|q_t - \mathbb{E}[q_t|\mathcal{H}_t]| \leq \epsilon_q$ with high probability.

Lemma C.10 (Approximate Adaptive Submodularity)**.** *Under assumptions (A3)-(A5), the quality-aware objective* $F(V) = \sum_k \mathcal{L}_{LM}(V; D_k) + \lambda_Q Q(V)$ *satisfies* ϵ*-approximate adaptive submodularity:*

$$\Delta_F((a,b)|\psi) \geq \Delta_F((a,b)|\psi') - \epsilon_{sub} \tag{6}$$

for $\psi \preceq \psi'$*, where* $\epsilon_{sub} = O(\epsilon_q + 1/\delta)$.

Proof sketch. The frequency-based component $\text{PMI}(a,b)$ exhibits exact diminishing returns: as more merges are performed, pair frequencies decrease, reducing potential PMI gains. The quality component $(\bar{q}_{ab})^\alpha$ is history-independent under (A3) and stable under (A5). The approximation error ϵ_{sub} arises from: (i) quality estimation noise (ϵ_q), and (ii) candidate pool variability ($1/\delta$). Full proof follows the framework of Golovin & Krause (2011). □

Theorem C.11 (Approximation Guarantee with Explicit Constants)**.** *Under Definition C.9, if assumptions (A1)-(A2) hold exactly, the greedy policy that maximizes* w_{ab} *achieves:*

$$F(\pi_{greedy}) \geq \left(1 - \frac{1}{e}\right) F(\pi^*) - K\epsilon_q - \frac{K}{\delta}, \tag{7}$$

where π^* *is the optimal adaptive policy over budget* K*. The error terms arise from* ϵ*-approximate submodularity (Lemma C.10).*

Proof. By Theorem 5 of Golovin & Krause (2011), greedy optimization of adaptive submodular functions achieves $(1-1/e)$ approximation. We extend this to ϵ-approximate submodularity (Lemma C.10).

With ϵ-approximate submodularity, the greedy per-step guarantee becomes $\Delta_F((a_t,b_t)|\psi_t) \geq \frac{1}{K}[F(\pi^*) - F(\psi_t)] - \epsilon_{\text{sub}}$. Defining $\Delta_t = F(\pi^*) - F_t$ and iterating over K steps: $\Delta_K \leq (1-1/K)^K \Delta_0 + K\epsilon_{\text{sub}} \leq \frac{1}{e}F(\pi^*) + K\epsilon_{\text{sub}}$, using $(1-1/K)^K \leq 1/e$.

Substituting $\epsilon_{\text{sub}} = \epsilon_q + 1/\delta$ yields $F(\pi_{\text{greedy}}) \geq (1-1/e)F(\pi^*) - K\epsilon_q - K/\delta$. □

Remark (Assumptions and Robustness): Assumptions (A1)–(A2) (adaptive monotonicity and submodularity) are **sufficient conditions** for the $(1-1/e)$ guarantee but may not hold exactly in practice. Specifically:

- The LM loss $\mathcal{L}_{\text{LM}}$ is not generally submodular in merge operations; the guarantee applies to the quality-frequency component $F(V)$ as defined.
- When assumptions are violated, the bound becomes approximate: $F(\pi_{\text{greedy}}) \geq (1-1/e)F(\pi^*) - K\epsilon_q - K/\delta - \epsilon_{\text{violation}}$, where $\epsilon_{\text{violation}}$ is proportional to the degree of assumption violation.

Empirically, our experiments show the guarantee is meaningful because: (1) tokenization objectives often exhibit near-submodular behavior (Lin & Bilmes, 2011); (2) end-to-end learning of α compensates for violations by calibrating the quality-frequency trade-off; (3) RL policy exploration in Stage 1 helps escape poor local optima that pure greedy would converge to.

C.8. Robustness Analysis

We analyze robustness under misspecified quality metrics and adversarial quality scores, quantifying interaction effects between RL and adaptive learning stages.

Theorem C.12 (Robustness to Quality Corruption)**.** *Let* $\tilde{q} = q + \xi$ *with* $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$*. Then*

$$\mathcal{L}(\tilde{q}) - \mathcal{L}(q) \leq \alpha\, \sigma_\xi \sqrt{\mathbb{E}[\|\nabla_q \mathcal{L}\|^2]}. \tag{8}$$

Proof. The result follows from Lipschitz stability of the bilevel objective. By the chain rule and Cauchy-Schwarz, $|\mathcal{L}(\tilde{q}) - \mathcal{L}(q)| \leq \|\xi\|_2 \cdot \int_0^1 \|\nabla_q \mathcal{L}(q + t\xi)\|_2 dt$.

From Proposition C.4, $\|\partial w_{ab}/\partial q\| \leq \alpha \epsilon_Q^{\alpha-1}$, yielding $\|\nabla_q \mathcal{L}\| \leq \alpha C_{\mathcal{L}}$. Taking expectations over $\xi \sim \mathcal{N}(0, \sigma_\xi^2 I)$ with $\mathbb{E}[\|\xi\|_2] \leq \sigma_\xi \sqrt{d}$ gives the stated bound. □

Empirical Validation. We validate the robustness bound experimentally:

- **20% quality noise**: Performance degradation of -4.2% (genomics) and -5.8% (finance), consistent with the $O(\alpha\sigma_\xi)$ bound.
- **Adversarial quality (inverted scores)**: Performance matches standard BPE, as expected when quality signals become uninformative.
- **50% missing quality**: Graceful fallback to frequency-only merging via the adaptive $\alpha \to 0$ mechanism.

Interaction Effects. We quantify the contribution of each learning stage:

- RL policy optimization alone: 65% of total improvement over BPE
- Adaptive parameter learning alone: 45% of total improvement
- Combined (synergy): Additional +10% from joint optimization

The super-additive effect (65% + 45% $>$ 100% total) indicates that the two stages reinforce each other: RL discovers promising merge patterns that adaptive learning then calibrates, while learned parameters improve the reward landscape for RL exploration.

C.9. Information-Theoretic Optimality

This subsection establishes that QA-Token achieves information-theoretic optimality under noisy conditions, providing theoretical justification for quality-aware tokenization.

Proposition C.13 (Quality-Aware Information Bottleneck Interpretation)**.** *Let X denote the input sequence, T the tokenized representation, and Y the downstream task labels. Under the quality-aware tokenization framework with quality scores Q, the optimal vocabulary V^* minimizes:*

$$\mathcal{L}_{QA}(V) = -I(T; Y|Q) + \beta \cdot I(T; X|Q) \tag{9}$$

where $I(\cdot;\cdot|\cdot)$ denotes conditional mutual information and β controls the compression-relevance tradeoff.

Connection to merge score: *The merge score w_{ab} is **consistent with** (but not directly derived from) this IB objective. PMI approximates compression efficiency $I(T; X|Q)$, while quality weighting ensures high $I(T; Y|Q)$ in reliable regions. The connection is qualitative rather than via direct differentiation of mutual information, which is intractable.*

Proof. Following the information bottleneck framework (Tishby et al., 1999) and its variational extension (Alemi et al., 2017), conditioning on quality Q yields the objective $\mathcal{L}_{\text{QA}} = -I(T; Y|Q) + \beta I(T; X|Q)$.

The merge score $w_{ab} \propto \text{PMI}(a, b) \cdot (\bar{q}_{ab})^\alpha$ is *consistent with* this IB objective: (i) the PMI term approximates compression efficiency $I(T; X|Q)$ (high-PMI merges compress efficiently), and (ii) the quality term weights merges by reliability, prioritizing high-quality regions for $I(T; Y|Q)$.

Caveat: The exact form of w_{ab} does not follow from direct differentiation of mutual information (intractable). Rather, it is a principled heuristic with end-to-end learning of α calibrating the quality-compression trade-off. □

Corollary C.14 (Noise Reduction Bound)**.** *For a corpus with noise level ϵ and quality scores q satisfying $\mathbb{E}[q|\textit{noise}] < \mathbb{E}[q|\textit{signal}]$, the quality-aware tokenizer achieves:*

$$\mathcal{L}_{QA} \leq \mathcal{L}_{\textit{uniform}} - \alpha \cdot \textit{Var}(q) \cdot \rho(q, \epsilon)^2 \tag{10}$$

where $\rho(q, \epsilon)$ is the correlation between quality scores and noise levels.

Key Theoretical Insights. This information-theoretic analysis provides three fundamental insights:

1. **Automatic Noise Filtering:** QA-Token implicitly performs importance sampling, up-weighting high-quality regions during vocabulary construction.
2. **Optimal Compression:** The quality-aware merge process achieves better rate-distortion tradeoffs by allocating more representation capacity to high-quality, informative regions.
3. **Transfer Learning:** Foundation models trained with QA-Token vocabularies learn more robust representations that transfer better to downstream tasks.

D. Complete Quality Metrics Formulations

D.1. Genomics: Detailed Sequencing Quality Metrics

In genomic sequencing, each nucleotide base call $s_i \in \{\text{A}, \text{C}, \text{G}, \text{T}, \text{N}\}$ is associated with a Phred quality score $Q_{\text{phred},i} \in [0, 93]$:

$$P_{\text{error}}(i) = 10^{-Q_{\text{phred},i}/10} \tag{11}$$

The base quality score is $q_i = 1 - P_{\text{error}}(i) \in [0, 1]$. Position-adjusted quality accounts for systematic degradation at read ends:

$$q'_i = q_i \cdot \exp\left(-\beta_{\text{pos}} \cdot \frac{|i - (L-1)/2|}{(L-1)/2 + \epsilon_{\text{len}}}\right) \tag{12}$$

where L is read length, $\beta_{\text{pos}} \geq 0$ is learnable, and $\epsilon_{\text{len}} = 10^{-6}$.

For multi-base token $t = s_1 ... s_{|t|}$, we use geometric mean aggregation:

$$q_t^{\text{genomic}} = \left(\prod_{j=1}^{|t|} q'_{s_j}\right)^{1/|t|} = \exp\left(\frac{1}{|t|}\sum_{j=1}^{|t|} \log(q'_{s_j} + \epsilon_Q)\right) \tag{13}$$

D.2. Finance: Comprehensive Market Quality Metrics

Financial time series quality combines four dimensions:

$$q_i^{\text{finance}} = \sum_{k=1}^{4} w_k \cdot q_{k,i}, \quad \sum_{k=1}^{4} w_k = 1 \tag{14}$$

1. Liquidity Quality:

$$q_{\text{liq}}(t) = \text{sigmoid}\left(\frac{\log(\text{volume}_t / \text{median\_volume})}{\sigma_{\text{volume}}}\right) \tag{15}$$

where σ_{volume} is the rolling standard deviation of log-volume computed over a lookback window of $L_{\text{vol}} = 252$ trading days (one year), clipped to $[0.1, 10]$ for numerical stability. This normalization ensures that q_{liq} responds proportionally to volume deviations relative to typical market activity.

2. Signal Quality:

$$q_{\text{sig}}(t) = \max\left(0, 1 - \frac{|\text{bid-ask spread}_t|}{\text{mid-price}_t \cdot \alpha_{\text{spread}}}\right) \tag{16}$$

3. Stability Quality:

$$q_{\text{stb}}(t) = \exp\left(-\beta_{\text{vol}} \cdot \frac{\text{realized\_vol}_t}{\text{expected\_vol}_t}\right) \tag{17}$$

where expected_vol_t is the exponentially weighted moving average (EWMA) of realized volatility following the RiskMetrics methodology (J.P. Morgan, 1996): $\text{expected\_vol}_t = \gamma_{\text{vol}} \cdot \text{expected\_vol}_{t-1} + (1 - \gamma_{\text{vol}}) \cdot \text{realized\_vol}_{t-1}$, with decay factor $\gamma_{\text{vol}} = 0.94$. The learnable parameter $\beta_{\text{vol}} \geq 0$ controls sensitivity to volatility spikes.

4. Information Quality:

$$q_{\text{info}}(t) = \frac{\text{MI}(\text{token}_t, \text{future\_return}_{t+h})}{\text{H}(\text{future\_return}_{t+h})} \tag{18}$$

Token aggregation uses arithmetic mean:

$$q_t^{\text{finance}} = \frac{1}{|t|} \sum_{i \in t} q_i^{\text{finance}} \tag{19}$$

Rationale for Arithmetic Mean Aggregation: Unlike genomics (which uses geometric mean, Eq. 13), financial data aggregation employs the arithmetic mean for two principled reasons: (1) *Additive noise model*: Financial market microstructure noise is predominantly additive across time points—a single noisy tick does not invalidate adjacent observations in the way a single low-quality DNA base compromises an entire read. Empirically, LOB noise sources (latency, partial fills, stale quotes) contribute independently rather than multiplicatively. (2) *Temporal continuity for forecasting*: Financial tokens represent contiguous time windows where downstream tasks (price prediction, volatility forecasting) operate on windowed features. The aggregate quality naturally represents the *average* reliability of observations within the window, which aligns with how prediction models weight inputs. In contrast, genomic tokens represent molecular sequences where any unreliable base compromises biological interpretation (e.g., variant calling), necessitating the conservative geometric mean that penalizes even single low-quality elements.

D.3. Social Media: Linguistic Quality Metrics

Social media text presents unique quality challenges including orthographic variations, semantic drift, platform-specific conventions, and temporal dynamics. We define a multi-dimensional quality vector for character-level tokens:

$$\mathbf{q}_t^{\text{social}} = (q_{\text{orth}}(t), q_{\text{sem}}(t), q_{\text{temp}}(t), q_{\text{plat}}(t)) \tag{20}$$

The scalar quality is obtained via learnable weighted aggregation:

$$q_t^{\text{social}} = \sum_j w_j \cdot q_j(t), \quad w_j \in \theta_{\text{adapt}} \tag{21}$$

1. Orthographic Quality: Measures deviation from canonical spelling:

$$q_{\text{orth}}(t) = \exp(-\lambda_{\text{edit}} \cdot d_{\text{edit}}(t, t_{\text{canonical}})) \tag{22}$$

where d_{edit} is the normalized Levenshtein distance to the nearest canonical form in a reference dictionary. The reference dictionary is constructed by combining: (i) the Hunspell en_US dictionary (2023 release, ≈140k entries), (ii) a curated social media slang lexicon (≈50k terms aggregated from NoSlang.com and similar sources), and (iii) domain-specific terminology lists for each benchmark task.

2. Semantic Quality: Captures contextual coherence:

$$q_{\text{sem}}(t) = \max(0, \cos(\vec{v}_t, \vec{v}_{\text{context}})) \tag{23}$$

where $\vec{v}_{\text{context}}$ is the average embedding of surrounding tokens. For efficiency, we use fastText Common Crawl embeddings (cc.en.300.bin, 2M vocabulary) (Bojanowski et al., 2017). For BERT-based variants requiring subword handling, we use `bert-base-uncased` from HuggingFace with mean pooling over subword tokens.

3. Temporal Quality: Models relevance decay over time:

$$q_{\text{temp}}(t) = \exp(-\gamma_{\text{decay}} \cdot \Delta t) \tag{24}$$

with time difference Δt in days from posting time, capturing trending topics and temporal relevance.

4. Platform Quality: Platform-specific noise modeling:

$$q_{\text{plat}}(t) = P(t|\text{platform}) \tag{25}$$

computed using 3-gram Kneser-Ney language models trained with KenLM (Heafield, 2011) on curated platform-specific corpora (≈10M tokens each): Twitter (tweets with $>$100 likes and $<$5% special characters), Reddit (comments with $>$10 upvotes from default subreddits), and Facebook (public posts from verified pages). These "clean" subsets establish platform-specific baselines for typical language patterns.

Learned Parameters: For the TweetEval benchmark experiments, the learned parameters were: $w_{\text{orth}} = 0.32 \pm 0.03$, $w_{\text{sem}} = 0.35 \pm 0.04$, $w_{\text{temp}} = 0.18 \pm 0.02$, $w_{\text{plat}} = 0.15 \pm 0.02$, $\lambda_{\text{edit}} = 0.5$, and $\gamma_{\text{decay}} = 0.01$.

E. Sequential Learning Process: Complete Framework

This section provides detailed algorithms and convergence analysis for QA-Token's two-stage sequential learning process.

E.1. Stage 1: Reinforcement Learning Policy Optimization

E.1.1. MDP Formulation

The vocabulary construction process is formulated as a finite-horizon Markov Decision Process (see Section E.7 for complete specification):

- **States $s_t \in \mathcal{S}$:** Encode current vocabulary V_t, merge candidates, corpus statistics, and progress t/T
- **Actions $a_t \in \mathcal{A}_t$:** Select a merge pair (a_i, b_i) from the priority queue
- **Transitions:** Deterministic vocabulary updates following merge operations
- **Rewards:** Multi-objective reward combining quality, information, and complexity

E.1.2. Reward Function Design

The reward function guides the RL agent:

$$R(a, b; \theta^{(0)}_{\text{adapt}}) = \sum_{j \in \{Q, I, C, \text{domain}\}} \lambda_j \hat{R}_j(a, b) \tag{26}$$

where components are normalized via exponential moving averages (see Section E.8). The detailed components are:

- **Quality Reward ($\hat{R}_Q$ from R^{raw}_Q):** Encourages high intrinsic quality for $t_{\text{merged}} = ab$, computed using domain-specific aggregation (Section D).
- **Information Reward ($\hat{R}_I$ from R^{raw}_I):** Rewards statistically significant merges, e.g., $R^{\text{raw}}_I(a, b) = \log \frac{P(t_{\text{merged}})}{P(a)P(b)+\epsilon_p}$.
- **Complexity Penalty ($\hat{R}_C$ from R^{raw}_C):** Typically negative, e.g., $R^{\text{raw}}_C(a, b) = -(|t_{\text{merged}}| \cdot \log(|V_t| + 1))$. $\hat{R}_C$ is then scaled to e.g. $[-1, 0]$.
- **Domain-Specific Rewards ($\hat{R}_{\text{domain},k}$ from $R^{\text{raw}}_{\text{domain},k}$):** Include conservation scores (genomics) and predictive power (finance).

The EMA-normalized rewards $\hat{R}_j(a, b)$ are used by the RL agent in Stage 1. For the Gumbel-Softmax logits in Stage 2 (Section E.9), raw or batch-normalized reward components are used to ensure direct differentiability with respect to θ_{adapt}.

E.1.3. PPO TRAINING ALGORITHM

Algorithm 1 Stage 1: RL Policy Training

Input: Corpus $\mathcal{S}$, initial $\theta^{(0)}_{\text{adapt}}$, episodes E
Initialize policy network π_{θ_π} and value network V_ϕ
for episode $e = 1$ to E **do**
  Initialize vocabulary $V_0 = \Sigma$
  for step $t = 1$ to T **do**
    Compute state features s_t from current vocabulary
    Sample action $a_t \sim \pi_{\theta_\pi}(a|s_t)$
    Execute merge $(a_{a_t}, b_{a_t}) \mapsto ab$
    Compute reward $r_t = R(a_{a_t}, b_{a_t}; \theta^{(0)}_{\text{adapt}})$
    Store trajectory (s_t, a_t, r_t)
  end for
  Update policy using PPO objective:
    $L^{\text{PPO}} = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$
  Update value network to minimize MSE
end for
Output: Optimized policy $\pi^*_{\theta_\pi}$

E.2. Stage 2: Adaptive Parameter Learning

E.2.1. ADAPTIVE PARAMETERS DEFINITION

The learnable parameter vector $\theta_{\text{adapt}} \in \mathbb{R}^m$ includes:

- **Quality sensitivity:** $\alpha \in [0, 2]$ controlling quality influence
- **Domain factors:** β_{pos} (genomics position decay), β_{vol} (finance volatility)
- **Quality weights:** $\mathbf{w} = (w_1, \ldots, w_k)$ for composite quality metrics
- **Reward weights:** $\boldsymbol{\lambda} = (\lambda_Q, \lambda_I, \lambda_C, \ldots)$ for multi-objective rewards

E.2.2. GUMBEL-SOFTMAX DIFFERENTIABLE OPTIMIZATION

To enable gradient-based optimization through discrete merge decisions, we employ Gumbel-Softmax relaxation:

Algorithm 2 Stage 2: Adaptive Parameter Learning

Input: Downstream dataset $\mathcal{D}$, policy $\pi^*_{\theta_\pi}$, initial θ_{adapt}
Initialize temperature $\tau = \tau_{\text{init}}$
for iteration $i = 1$ to N **do**
  Sample batch B from $\mathcal{D}$
  for each sequence in batch **do**
    Generate top-K merge candidates via priority queue ranked by w_{ab}
    Compute composite logits: $\ell_{ab} = w_{ab}(a, b; \alpha) + \sum_j \lambda_j R_j^{\text{raw}}$
    Select merge via Gumbel-Softmax (differentiable relaxation):
      $y_i = \frac{\exp((\ell_i + g_i)/\tau)}{\sum_j \exp((\ell_j + g_j)/\tau)}$
    Construct differentiable tokenized representation
  end for
  Compute task loss L_{task} on tokenized batch
  Update parameters: $\theta_{\text{adapt}} \leftarrow \theta_{\text{adapt}} - \eta \nabla L_{\text{total}}$
  Anneal temperature: $\tau \leftarrow \tau \cdot \exp(-\beta_{\text{anneal}})$
end for
Output: Optimized parameters θ^*_{adapt}

E.3. Final Vocabulary Construction

After completing both stages, the final vocabulary for deployment is constructed.

Detailed Process: Following the completion of Stage 1 (RL policy optimization yielding $\pi^*_{\theta_\pi}$) and Stage 2 (adaptive parameter learning yielding θ^*_{adapt}), the final vocabulary for deployment is typically constructed. While several strategies are possible, our primary approach involves the optimized adaptive parameters θ^*_{adapt} to re-evaluate merge priorities. Specifically, a greedy BPE-like process is executed, starting from the base alphabet. At each step, the merge operation (a, b) is chosen that maximizes the quality-aware merge score $w_{ab}(a, b; \theta^*_{\text{adapt}})$ as defined in Equation 5, using the learned parameters within θ^*_{adapt} (e.g., α^*). This process continues until the target vocabulary size is reached. Alternatively, if the RL policy $\pi^*_{\theta_\pi}$ is robust across variations in θ_{adapt}, it could be used with inputs (state features, merge scores) calculated using θ^*_{adapt}. However, the greedy approach based on $w_{ab}(\theta^*_{\text{adapt}})$ is generally more direct and computationally efficient for deployment, leveraging the refined understanding of "good" merges embodied in θ^*_{adapt}.

Algorithm 3 Final Vocabulary Construction

Input: Corpus $\mathcal{S}$, optimized θ^*_{adapt}, target size K
Initialize vocabulary $V = \Sigma$, merge count $m = 0$
while $m < K$ **do**
  Compute all merge scores: $w_{ab} = \frac{f(a,b)}{f(a)f(b)+\epsilon_f} \cdot (\bar{q}_{ab} + \epsilon_Q)^{\alpha^*} \cdot \psi(a, b)$
  Select best merge: $(a^*, b^*) = \arg\max_{(a,b)} w_{ab}$
  Update vocabulary: $V \leftarrow V \cup \{a^* b^*\}$ // Constituents a^*, b^* remain in V
  Update corpus statistics and recompute affected frequencies
  $m \leftarrow m + 1$
end while
Output: Final vocabulary V^*

E.4. Convergence Properties

The sequential learning process has the following theoretical guarantees:

Theorem E.1 (Two-Timescale Convergence)**.** *Under assumptions A1-A4 (Section C.6), the sequential optimization of θ_π (fast timescale) and θ_{adapt} (slow timescale) converges to a local Nash equilibrium with probability 1.*

Proof. The result follows from two-timescale stochastic approximation (Borkar, 2009). Under (A1)–(A4), the conditions of Theorem 2 in (Borkar, 2009) are satisfied: (i) Lipschitz gradients (from bounded rewards and smooth parameterization),

(ii) bounded iterates via projection, (iii) martingale noise with bounded variance, and (iv) proper step sizes ($\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$).

With timescale separation $\eta_\pi^{(t)}/\eta_{\text{adapt}}^{(t)} \to \infty$, the fast iterate θ_π equilibrates before significant movement in θ_{adapt}. The iterates converge almost surely to limit points $(\theta_\pi^*, \theta_{\text{adapt}}^*)$ satisfying $\nabla_{\theta_\pi} J = 0$ and $\nabla_{\theta_{\text{adapt}}} L_{\text{total}} = 0$, constituting a local Nash equilibrium. □

Key Properties:

- **Stage 1 Convergence:** PPO converges to a stationary point at rate $O(1/\sqrt{T})$ (Proposition C.6)
- **Stage 2 Convergence:** Gumbel-Softmax optimization converges at rate $O(1/\sqrt{T}) + O(\tau^2)$ (Proposition C.7)
- **Overall Optimality:** The greedy vocabulary construction with θ_{adapt}^* achieves $(1 - 1/e)$-approximation (Theorem C.11)

Proposition E.2 (Convergence of Adaptive Learning with Explicit Constants)**.** *Under Assumptions A1–A4, with $\eta_t = \eta_0/\sqrt{t}$ and $\eta_0 \leq 1/(2L)$, where L is the Lipschitz constant of ∇L_{total}, we have:*

$$\mathbb{E}[\|\nabla L_{total}(\theta_{adapt}^T)\|^2] \leq \frac{2(L_{total}(\theta_{adapt}^0) - L^*)}{\eta_0\sqrt{T}} + \frac{4\eta_0 L\sigma^2}{\sqrt{T}}, \tag{27}$$

where L^* is the optimal value and σ^2 bounds gradient variance.

Proof. The proof follows standard non-convex SGD analysis (Kingma & Ba, 2014). By smoothness of L_{total}:

$$L_{\text{total}}(\theta^{t+1}) \leq L_{\text{total}}(\theta^t) - \eta_t\langle\nabla L_{\text{total}}(\theta^t), g_t\rangle + \frac{L\eta_t^2}{2}\|g_t\|^2 \tag{28}$$

where g_t is the stochastic gradient. Taking expectations and using $\mathbb{E}[g_t] = \nabla L_{\text{total}}(\theta^t)$ and $\mathbb{E}[\|g_t\|^2] \leq \|\nabla L_{\text{total}}(\theta^t)\|^2 + \sigma^2$:

$$\mathbb{E}[L_{\text{total}}(\theta^{t+1})] \leq \mathbb{E}[L_{\text{total}}(\theta^t)] - \eta_t(1 - L\eta_t/2)\mathbb{E}[\|\nabla L_{\text{total}}(\theta^t)\|^2] + \frac{L\eta_t^2\sigma^2}{2} \tag{29}$$

Telescoping over T iterations with $\eta_t = \eta_0/\sqrt{t}$ and $\eta_0 \leq 1/(2L)$ yields the stated bound.

Remark: Under temperature annealing $\tau_t \to 0$, the Gumbel-Softmax bias term $B(\tau)^2 \to 0$, ensuring asymptotic unbiasedness. □

Theorem E.3 (Local vs Global Optimality)**.** *The two-timescale optimization converges to a local Nash equilibrium $(\theta_\pi^*, \theta_{adapt}^*)$ with quality bounds under local strong convexity; probabilistic restarts increase the chance of reaching global optima.*

Proof. **Part 1: Local Nash Equilibrium.** By Theorem E.1, the limit points satisfy $\nabla_{\theta_\pi} J = 0$ and $\nabla_{\theta_{\text{adapt}}} L_{\text{total}} = 0$, constituting a local Nash equilibrium.

Part 2: Quality Bounds. Under μ-strong convexity of L_{total} in neighborhood $\mathcal{B}_r(\theta_{\text{adapt}}^*)$:

$$L_{\text{total}}(\theta_{\text{adapt}}^*) - L_{\text{total}}(\theta_{\text{adapt}}^{\text{global}}) \leq \frac{1}{2\mu}\|\nabla L_{\text{total}}(\theta_{\text{adapt}}^*)\|^2 = 0 \tag{30}$$

if the global optimum lies within the basin of attraction.

Part 3: Probabilistic Restarts. With M independent runs, $\mathbb{P}[\text{find global}] = 1 - (1 - p_{\text{basin}})^M \geq 1 - e^{-M \cdot p_{\text{basin}}}$, achieving probability $\geq 1 - \delta$ for $M = O(\log(1/\delta)/p_{\text{basin}})$ restarts. □

E.5. Algorithm Summary

Algorithm 4 QA-Token: Quality-Aware Tokenization Framework

Input: Corpus $\mathcal{C}$, quality scores Q, vocabulary budget K
Output: Optimized vocabulary V^*

Stage 1: RL Policy Optimization
Initialize policy π_{θ_π}, adaptive parameters $\theta^{(0)}_{\text{adapt}}$
for episode $e = 1$ to E **do**
  $V \leftarrow \Sigma$ (base alphabet)
  for step $t = 1$ to K **do**
    Compute priority queue PQ_t with scores $w_{ab}(\cdot; \theta^{(0)}_{\text{adapt}})$
    Select merge $(a, b) \sim \pi_{\theta_\pi}(\cdot|s_t)$ from PQ_t
    Execute merge: $V \leftarrow V \cup \{ab\}$ // Add merged token
    Compute reward R_t using Eq. 26
  end for
  Update π_{θ_π} via PPO using trajectory rewards
end for

Stage 2: Adaptive Parameter Learning
for iteration $i = 1$ to I **do**
  Sample mini-batch of merge candidates $\mathcal{B}$
  Compute logits $\ell_{ab}(\theta_{\text{adapt}})$ using Eq. 37
  Sample Gumbel noise and compute soft selection via Eq. 38
  Evaluate task loss L_{task} on downstream objective
  Update $\theta_{\text{adapt}} \leftarrow \theta_{\text{adapt}} - \eta_i \nabla L_{\text{total}}$
end for

Final Vocabulary Construction
Build final vocabulary using greedy merges with $w_{ab}(\cdot; \theta^*_{\text{adapt}})$
Return V^*

Algorithm 5 QA-Token Integration with Downstream Transformer

Input: Raw sequence X, trained QA-Token vocab V^*, Transformer model M_θ
Output: Task predictions $\hat{Y}$

// Tokenization (no overhead vs. BPE)
$T \leftarrow \text{Tokenize}(X, V^*)$ // Standard greedy tokenization

// Embedding and Encoding
$E \leftarrow \text{TokenEmbed}(T) + \text{PosEmbed}(\text{positions})$
for layer $\ell = 1$ to L **do**
  $E \leftarrow \text{TransformerBlock}_\ell(E)$
end for

// Task Head
$\hat{Y} \leftarrow \text{TaskHead}(E)$ // Classification, regression, or generation
Return $\hat{Y}$

Algorithm 6 Meta-Learning Initialization for Adaptive Parameters

Require: Task distribution $\mathcal{P}(\mathcal{T})$, base initialization $\theta^{(0)}_{\text{adapt}}$, inner steps K, inner lr η_{in}, outer lr η_{out}
while not converged **do**
  Sample batch of tasks $\{\mathcal{T}_i\} \sim \mathcal{P}(\mathcal{T})$
  for each task $\mathcal{T}_i$ **do**
    Set $\theta_i \leftarrow \theta^{(0)}_{\text{adapt}}$
    for $k = 1 \dots K$ **do**
      Compute $L^{(i)}_{\text{total}}(\theta_i)$ on $\mathcal{T}_i$ and update $\theta_i \leftarrow \theta_i - \eta_{\text{in}} \nabla_\theta L^{(i)}_{\text{total}}(\theta_i)$
    end for
  end for
  Update initialization: $\theta^{(0)}_{\text{adapt}} \leftarrow \theta^{(0)}_{\text{adapt}} - \eta_{\text{out}} \sum_i \nabla_{\theta^{(0)}_{\text{adapt}}} L^{(i)}_{\text{total}}(\theta_i)$
end while

return meta-initialization $\theta^\star_{\text{adapt}}$

E.6. RL Training Implementation

E.6.1. State Representation

The state s_t provided to the RL agent at merge step t includes:

- **Global Features:** Current vocabulary size $|V_t|$; remaining merge steps $T - t$; aggregated token statistics (average length, mean/std of quality scores).
- **Candidate Pair Features (top-K_{PQ} from priority queue):** For each pair (a, b): frequencies $f(a), f(b), f(a, b)$; qualities q_a, q_b; lengths $|a|, |b|$; merge score w_{ab}.
- **Domain Context:** Market regime indicators (finance), platform ID (social media), or sequence quality (genomics).

The PPO agent uses an MLP with 2 hidden layers (256 units each, ReLU activations). The policy network outputs action probabilities over K_{PQ} candidates; the value network outputs a single scalar.

E.6.2. Exploration Strategy

An ϵ-greedy strategy is employed with ϵ annealed from $\epsilon_0 = 0.5$ to $\epsilon_{\text{final}} = 0.05$ over training episodes using exponential decay, balancing exploration and exploitation effectively across all domains.

E.7. MDP Formulation and Details

Definition E.4 (Tokenization MDP)**.** The tokenization MDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, T)$ where:

1. **State Space $\mathcal{S}$:** Each state $s_t \in \mathcal{S} \subset \mathbb{R}^d$ encodes:
   - Current vocabulary V_t and its statistics (size, token length distribution)
   - Priority queue $PQ_t = \{(a_i, b_i, w_{a_i b_i})\}_{i=1}^{K_{PQ}}$ of top merge candidates
   - Corpus statistics: frequency distributions, quality histograms
   - Progress indicator: t/T where T is the merge budget

   Formally, $s_t = [\phi(V_t), \phi(PQ_t), \phi(\mathcal{S}_t), t/T] \in \mathbb{R}^d$.

   State Encoding Function ϕ: The encoding function $\phi : \mathcal{X} \to \mathbb{R}^{d_\mathcal{X}}$ maps variable-size structures to fixed-dimensional vectors:
   - $\phi(V_t) = [|V_t|/|\Sigma|, \bar{|t|}, \sigma_{|t|}, \bar{q}_t, \sigma_{q_t}] \in \mathbb{R}^5$: vocabulary size ratio, mean/std of token lengths, mean/std of token qualities.

- $\phi(PQ_t) \in \mathbb{R}^{6 \cdot K_{PQ}}$: For top-$K_{PQ}$ candidates, concatenate $[w_{ab}, q_a, q_b, |a|, |b|, \log f(a,b)]$ per pair. Pad with zeros if fewer candidates exist.
- $\phi(\mathcal{S}_t) \in \mathbb{R}^{20}$: Quality histogram ($B_q = 10$ bins over $[0,1]$) and log-frequency histogram ($B_f = 10$ bins over observed range).

Total state dimension: $d = 5 + 6 \cdot K_{PQ} + 20 + 1$. With $K_{PQ} = 50$, we have $d = 326$. The MLP policy network processes this representation via two hidden layers (256, 128 units) with ReLU activations (see Appendix E.6).

2. **Action Space $\mathcal{A}_t$:** At time t:
$$\mathcal{A}_t = \{i : (a_i, b_i) \in PQ_t, i \leq K_{PQ}\} \tag{31}$$
Each action $a_t \in \mathcal{A}_t$ selects a merge pair from the priority queue.

3. **Transition Dynamics $\mathcal{P}$:** Deterministic transitions:
$$s_{t+1} = \mathcal{P}(s_t, a_t) = \text{UPDATE}(s_t, \text{MERGE}(a_{a_t}, b_{a_t})) \tag{32}$$
where MERGE executes vocabulary update and UPDATE recomputes statistics.

4. **Reward Function:** $\mathcal{R}(s_t, a_t) = R(a_{a_t}, b_{a_t}; \theta^{(0)}_{\text{adapt}})$

5. **Discount Factor:** $\gamma = 1$ (undiscounted, finite horizon)

6. **Horizon:** $T = K$ merge operations

Proposition E.5 (MDP Well-Formedness)**.** *The tokenization MDP satisfies:*

1. *Markov Property:* $P(s_{t+1}|s_t, a_t, s_{t-1}, \ldots) = P(s_{t+1}|s_t, a_t)$
2. *Bounded State Space:* $\|s_t\|_2 \leq C_s$
3. *Finite Action Space:* $|\mathcal{A}_t| \leq K_{PQ}$

Proof. (1) follows from state containing complete information for transitions. (2) holds as vocabulary size is bounded by $|\Sigma| + T$ and frequencies are normalized. (3) is by construction of the priority queue. □

□

E.8. Reward Normalization Details

Each raw reward component $R^{\text{raw}}_j(a,b)$ is normalized using adaptive running statistics. We maintain exponential moving averages (EMAs) for mean $\mu^{\text{run}}_{j,t}$ and variance $\text{Var}^{\text{run}}_{j,t}$:

$$\mu^{\text{run}}_{j,t} = (1 - \beta_{\text{norm}})\mu^{\text{run}}_{j,t-1} + \beta_{\text{norm}} R^{\text{raw}}_j(a,b) \tag{33}$$
$$\text{Var}^{\text{run}}_{j,t} = (1 - \beta_{\text{norm}})\text{Var}^{\text{run}}_{j,t-1} + \beta_{\text{norm}}(R^{\text{raw}}_j(a,b) - \mu^{\text{run}}_{j,t-1})(R^{\text{raw}}_j(a,b) - \mu^{\text{run}}_{j,t}) \tag{34}$$

where $\beta_{\text{norm}} \in [10^{-3}, 10^{-2}]$. The normalized component is:

$$\hat{R}_j(a,b) = \frac{R^{\text{raw}}_j(a,b) - \mu^{\text{run}}_{j,t-1}}{\sigma^{\text{run}}_{j,t-1} + \epsilon_R} \tag{35}$$

with $\epsilon_R = 10^{-8}$ for stability.

E.9. Gumbel-Softmax Gradient Derivation and Temperature Annealing

E.10. Temperature Annealing Schedule

We employ an exponential annealing schedule for the temperature parameter:

$$\tau(t) = \tau_{\text{init}} \cdot \exp(-\beta_{\text{anneal}} \cdot t/T_{\text{anneal}}), \tag{36}$$

where $\tau_{\text{init}} = 1.0$, $\beta_{\text{anneal}} = 3.0$, and T_{anneal} is the total number of optimization steps.

This schedule ensures:

- **Early exploration:** High initial temperature allows exploration of diverse merge patterns
- **Gradual refinement:** Exponential decay provides smooth transition to discrete selections
- **Convergence:** Low final temperature approaches one-hot categorical sampling

E.11. Gradient Computation

The composite logits for candidate merge (a, b) are:

$$\ell_{ab}(\theta_{\text{adapt}}) = w_{ab}(a, b; \alpha) + \sum_j \lambda_j R_j^{\text{raw}}(a, b), \tag{37}$$

which are differentiable with respect to θ_{adapt} through both the merge score and reward weights.

The composite logits combine w_{ab} (which incorporates frequency via PMI and quality via $\bar{q}_{ab}$) with raw reward components R_j^{raw} that also capture quality (R_Q^{raw}) and information (R_I^{raw}).

Rationale for Intentional Overlap: While there is deliberate overlap between these components (both encode quality and frequency signals), they serve *distinct optimization purposes*:

- w_{ab} **(merge score):** Optimized via the RL objective (cumulative discounted reward) during Stage 1, capturing *corpus-level* quality-frequency tradeoffs that generalize across merge sequences.
- $\sum_j \lambda_j R_j^{\text{raw}}$ **(weighted rewards):** Optimized via the downstream task loss L_{task} during Stage 2, enabling *task-specific* reweighting of quality vs. information vs. complexity.

This decomposition allows the system to learn *different* quality-frequency tradeoffs for policy learning (Stage 1) versus task-specific adaptation (Stage 2). The parameter α controls general token quality preferences learned from reward maximization, while λ_j adjusts relative importance based on task-specific gradients. Ablation studies (Appendix G.5, Table 13) confirm that removing either component degrades downstream performance by 3–8%, validating that both contributions are necessary despite their overlap.

The Gumbel-Softmax distribution provides a differentiable approximation:

$$y_i = \frac{\exp((\ell_i + g_i)/\tau)}{\sum_{j=1}^{|\mathcal{C}|} \exp((\ell_j + g_j)/\tau)}, \quad g_i \sim \text{Gumbel}(0, 1) \tag{38}$$

The gradient of the task loss is computed via Monte Carlo sampling:

$$\nabla_{\theta_{\text{adapt}}} L_{\text{task}} = \mathbb{E}_{\mathbf{g}} \left[\nabla_{\theta_{\text{adapt}}} L_{\text{task}}(\mathbf{y}(\boldsymbol{\ell}(\theta_{\text{adapt}}), \mathbf{g}, \tau)) \right] \tag{39}$$

where $\mathbf{g}$ is sampled Gumbel noise.

Gradient Flow: The gradient flows through:

1. **Task loss:** L_{task} evaluates performance on downstream data
2. **Soft tokenization:** Gumbel-Softmax provides differentiable token boundaries
3. **Merge logits:** ℓ_{ab} depends on learnable θ_{adapt}
4. **Quality scores:** Through α and domain parameters $\beta_{\text{pos}}, \beta_{\text{vol}}$
5. **Reward weights:** Through $\boldsymbol{\lambda}$ in the composite score

F. Hyperparameter Sensitivity Analysis

We conducted a comprehensive sensitivity analysis on key parameters of the QA-Token framework: the quality sensitivity exponent α, the primary quality reward weight λ_Q, and the domain-specific volatility scaling exponent β_{vol} for finance. For each parameter, we varied its value across a specified range while holding all other hyperparameters at their optimal values, as determined during the adaptive learning phase.

The results, summarized in Table 10, demonstrate that while performance is optimal at the learned parameter values, the framework is not unduly sensitive to minor perturbations. Performance degrades gracefully rather than catastrophically as parameters deviate from their optima, suggesting the model occupies a reasonably wide basin of attraction in the hyperparameter space.

Table 10. Hyperparameter Sensitivity Analysis. Performance on the primary metric is reported as key hyperparameters are varied around their learned optimal value (indicated by *). Values are means over $n = 5$ runs.

Parameter	Value	Performance Metric
Genomics (QA-BPE-seq) - Metric: Variant F1		
α (Quality Sensitivity)	0.3	0.869
	0.5	0.879
	0.72*	**0.891**
	1.0	0.884
	1.5	0.872
λ_Q (Quality Reward Weight)	0.15	0.879
	0.25	0.886
	0.35*	**0.891**
	0.45	0.885
	0.55	0.878
Finance (QAT-QF) - Metric: Sharpe Ratio		
α (Quality Sensitivity)	0.25	1.61
	0.50	1.68
	0.95*	**1.72**
	1.50	1.65
	2.00	1.58
β_{vol} (Volatility Scaling)	0.10	1.63
	0.30	1.69
	0.50*	**1.72**
	0.70	1.67
	1.00	1.60

G. Complete Experimental Results

This section provides comprehensive experimental results across all domains, including detailed analysis, foundation-scale evaluations, and ablation studies.

G.1. Genomics Results: Detailed Analysis

Key Observations: QA-BPE-seq achieves 4.0 percentage point F1 improvement in variant calling over DNABERT-k (0.891 vs. 0.851) with Hedges' $g = 8.2$—a large effect size. Compared to standard BPE (0.824), the improvement is 6.7 percentage points. Taxonomic classification shows 3.1 percentage point gain over domain-standard k-mer tokenization. Sequence reconstruction improves by 16%, indicating information preservation.

Key Insights:

1. **Byte-level models fail catastrophically:** ByT5 and CANINE show 2.5× slower inference with 7-9% lower accuracy, definitively establishing that vocabulary-based tokenization remains essential for genomic sequences.
2. **Quality awareness is learnable:** The converged parameters ($\alpha = 0.72 \pm 0.03$, $\beta_{\text{pos}} = 0.014 \pm 0.002$) demonstrate that optimal quality sensitivity can be discovered through our adaptive learning framework.
3. **Mechanism of improvement:** Analysis of generated vocabularies reveals that QA-BPE-seq creates tokens aligned with biological units (codons, motifs) while breaking at error-prone junctions—a behavior that emerges without explicit biological supervision.

G.2. Financial Foundation Model: Detailed Results Analysis

QAT-QF demonstrates remarkable consistency across all financial tasks, with zero-shot improvements ranging from 7.3% to 27.0

Specific Observations:

- The model's superior performance on regime detection (+11.6% F1) and tail risk estimation (+18.0%) suggests that quality-aware tokenization captures market dynamics that frequency-based methods miss.
- Particularly noteworthy is the 27.0% improvement in order flow imbalance prediction, a task highly sensitive to microstructure noise.
- These results validate our hypothesis that incorporating quality signals during tokenization enables foundation models to learn more robust representations of financial time series.

G.3. Computational Costs

Training Time.

- Standard BPE: 5–10 minutes (5GB, CPU)
- QA-Token Stage 1 (RL): 30–36 GPU-hours (A100)
- QA-Token Stage 2 (Adaptive): 20–24 GPU-hours

Memory Requirements.

- Priority Queue: $O(K_{PQ} \cdot d)$ ($\sim$10 MB for K_{PQ}=200)
- Quality Statistics: $O(|V| \cdot s)$ ($\sim$100 MB for 32K vocab)
- Pair Frequencies: $O(|V|^2)$ ($\sim$4 GB for 32K vocab)
- Peak: $\sim$16 GB GPU

Hierarchical Training via Quality-Stratified Sampling. For massive corpora where full vocabulary optimization is infeasible, we employ *quality-variance importance sampling*: data points are sampled with probability proportional to $\text{Var}(q_i) + \epsilon_{\text{base}}$, prioritizing regions with heterogeneous quality where tokenization decisions have the greatest impact.

Definition G.1 (Notation for Hierarchical Training)**.** Let $\mathcal{C}$ denote the full corpus and $\mathcal{S} \subseteq \mathcal{C}$ a subset with $|\mathcal{S}| = r \cdot |\mathcal{C}|$ for *subset ratio* $r \in (0, 1]$. Define:

- $\mathcal{L}(V; D) = \mathbb{E}_{x \sim D}[-\log P_{\text{LM}}(x|V)]$: expected language modeling loss on distribution D using vocabulary V
- $V_{\mathcal{S}}^* = \arg\min_V \mathcal{L}(V; \mathcal{S})$: optimal vocabulary for subset $\mathcal{S}$
- $V_{\mathcal{C}}^* = \arg\min_V \mathcal{L}(V; \mathcal{C})$: optimal vocabulary for full corpus $\mathcal{C}$

Proposition G.2 (Hierarchical Training Bound)**.** *Under the following assumptions:*

(A1) The loss $\mathcal{L}(V;\cdot)$ is L-Lipschitz in the data distribution (bounded sensitivity to distribution shift)

(A2) Quality-variance importance sampling achieves effective sample size $n_{eff} = r \cdot |\mathcal{C}|/(1+CV^2)$ where CV is the coefficient of variation of importance weights

Then the vocabulary $V_{\mathcal{S}}^*$ learned on the importance-sampled subset satisfies:

$$\mathbb{E}[\mathcal{L}(V_{\mathcal{S}}^*;\mathcal{C})] \leq \mathcal{L}(V_{\mathcal{C}}^*;\mathcal{C}) + O\left(L \cdot \sqrt{\frac{1+CV^2}{r \cdot |\mathcal{C}|}}\right). \tag{40}$$

Proof Sketch. The bound follows from standard importance sampling theory (Owen, 2013). Under (A1), the difference $|\mathcal{L}(V;\mathcal{S}) - \mathcal{L}(V;\mathcal{C})|$ is controlled by the distributional divergence between $\mathcal{S}$ and $\mathcal{C}$. Importance sampling with weights $w_i \propto \text{Var}(q_i) + \epsilon_{\text{base}}$ reduces this divergence by oversampling high-variance regions where tokenization quality matters most. By the effective sample size formula for importance sampling, the estimation error scales as $O(1/\sqrt{n_{\text{eff}}})$, yielding the stated bound. The Lipschitz assumption (A1) ensures that optimization on $\mathcal{S}$ transfers to $\mathcal{C}$. □

Massive-Scale Strategies (>100TB).

1. Quality-stratified sampling (0.1–1%)
2. Distributed PPO (8–32 GPUs)
3. Online RL with replay for streams
4. Memory-mapped frequency tables

Cost-Benefit.

- +5–30% task performance
- -15–20% token count (faster inference)
- One-time cost amortized across applications

G.4. Foundation-Scale Results (Pathogen Detection, GUE)

Table 11. Pathogen Detection benchmark (MCC).

Task	DNABERT-2	DNABERT-S	NT-2.5b-Multi	NT-2.5b-1000g	METAGENE-1	METAGENE-1 (QA-Token)
Pathogen-Detect (avg.)	87.92	87.02	82.43	79.02	92.96	**94.53**
Pathogen-Detect-1	86.73	85.43	83.80	77.52	92.14	**93.81**
Pathogen-Detect-2	86.90	85.23	83.53	80.38	90.91	**92.95**
Pathogen-Detect-3	88.30	89.01	82.48	79.83	93.70	**95.12**
Pathogen-Detect-4	89.77	88.41	79.91	78.37	95.10	**96.24**

G.5. Ablation Studies and Additional Experiments

G.5.1. RL Algorithm Ablation

We assess the sensitivity of QA-Token to the choice of RL optimizer by replacing PPO with GRPO and VAPO (implementations following (Shao et al., 2024; Yue et al., 2025)). Across domains, downstream performance is stable and vocabulary similarity remains high (Jaccard ≥ 0.95), confirming modularity of the framework.

Table 12. Genome Understanding Evaluation (GUE). All metrics are MCC except COVID which uses F1.

Task	CNN	HyenaDNA	DNABERT	NT-2.5B-Multi	DNABERT-2	METAGENE-1	METAGENE-1 (QA-Token)
TF-Mouse (AVG.)	45.3	51.0	57.7	67.0	68.0	**71.4**	**72.8**
0	31.1	35.6	42.3	**63.3**	56.8	61.5	62.1
1	59.7	80.5	79.1	83.8	**84.8**	83.7	84.1
2	63.2	65.3	69.9	71.5	**79.3**	83.0	**84.5**
3	45.5	54.2	55.4	69.4	66.5	82.2	**83.3**
4	27.2	19.2	42.0	47.1	**52.7**	46.6	47.0
TF-HUMAN (AVG.)	50.7	56.0	64.4	62.6	**70.1**	68.3	69.9
0	54.0	62.3	68.0	66.6	**72.0**	68.9	70.2
1	63.2	67.9	70.9	66.6	**76.1**	70.8	72.0
2	45.2	46.9	60.5	58.7	**66.5**	65.9	66.8
3	29.8	41.8	53.0	51.7	**58.5**	58.1	59.0
4	61.5	61.2	69.8	69.3	77.4	77.9	**78.5**
EMP (AVG.)	37.6	44.9	49.5	58.1	56.0	66.0	**67.5**
H3	61.5	67.2	74.2	78.8	78.3	80.2	**81.0**
H3K14AC	29.7	32.0	42.1	56.2	52.6	64.9	**66.0**
H3K36ME3	38.6	48.3	48.5	62.0	56.9	66.7	**67.8**
H3K4ME1	26.1	35.8	43.0	**55.3**	50.5	55.3	**56.1**
H3K4ME2	25.8	25.8	31.3	36.5	31.1	51.2	**52.3**
H3K4ME3	20.5	23.1	28.9	40.3	36.3	58.5	**59.5**
H3K79ME3	46.3	54.1	60.1	64.7	67.4	73.0	**74.1**
H3K9AC	40.0	50.8	50.5	56.0	55.6	65.5	**66.5**
H4	62.3	73.7	78.3	81.7	80.7	82.7	**83.5**
H4AC	25.5	38.4	38.6	49.1	50.4	61.7	**62.8**
PD (AVG.)	77.1	35.0	84.6	**88.1**	84.2	82.3	85.5
ALL	75.8	47.4	90.4	**91.0**	86.8	86.0	88.5
NO-TATA	85.1	52.2	93.6	94.0	94.3	93.7	**94.5**
TATA	70.3	5.3	69.8	**79.4**	71.6	67.4	73.5
CPD (AVG.)	62.5	48.4	**73.0**	71.6	70.5	69.9	71.2
ALL	58.1	37.0	**70.9**	70.3	69.4	66.4	68.0
NO-TATA	60.1	35.4	69.8	**71.6**	68.0	68.3	69.5
TATA	69.3	72.9	**78.2**	73.0	74.2	75.1	76.3
SSD	76.8	72.7	84.1	89.3	85.0	87.8	**89.5**
COVID (F1)	22.2	23.3	62.2	73.0	71.9	72.5	**73.3**
GLOBAL WIN %	0.0	0.0	7.1	21.4	25.0	46.4	**57.1**

Table 13. Ablation across RL algorithms with training time (GPU-h), inference time (ms/seq), and vocabulary Jaccard similarity vs. PPO.

Domain	Config (Metric)	Metric Value	Train Time (GPU-h)	Inference (ms/seq)	Vocab Jaccard
Genomics	QA-Token (PPO)	**0.891**	34.0	10.2	1.00
	QA-Token (GRPO)	0.890	32.5	10.3	0.98
	QA-Token (VAPO)	0.892	31.8	10.2	0.97
	QA-Token (DAPO)	0.889	34.2	10.4	0.98
Finance	QA-Token (PPO)	**1.72**	28.0	15.2	1.00
	QA-Token (GRPO)	1.71	26.5	15.3	0.96
	QA-Token (VAPO)	1.73	25.0	15.1	0.95
	QA-Token (DAPO)	1.70	28.5	15.2	0.96
Social	QA-Token (PPO)	**74.5**	30.0	12.5	1.00
	QA-Token (GRPO)	74.2	29.0	12.6	0.97
	QA-Token (VAPO)	74.6	28.0	12.5	0.98
	QA-Token (DAPO)	74.3	31.0	12.7	0.97

G.6. Data Curation Baseline: BPE on Clean Data vs. QA-Token on Noisy Data

A natural question is whether simply filtering to high-quality data and using standard BPE could match QA-Token's performance. We evaluate this data curation baseline by training BPE on only the top 20% highest-quality sequences (average Phred score ≥ 30) and comparing against QA-Token trained on the full noisy corpus.

Table 14. Data Curation Baseline Comparison (Genomics Variant Calling). QA-Token on noisy data outperforms BPE on curated clean data.

Method	Training Data	Variant F1	p-value
BPE (full corpus)	100% (noisy)	0.824 ± 0.004	< 0.001
BPE (top 20% clean)	20% (Q$\geq$30)	0.847 ± 0.005	< 0.001
QA-Token	100% (noisy)	$\mathbf{0.891 \pm 0.004}$	—

Key findings:

- Data curation (BPE on clean data) improves over BPE on full noisy data: $+2.8\%$ F1 (0.847 vs 0.824).
- However, QA-Token on the *full noisy corpus* outperforms BPE on clean data by $+5.2\%$ F1 (0.891 vs 0.847, $p < 0.001$).
- This demonstrates that quality-aware tokenization extracts more value from noisy data than discarding it entirely.

G.7. Genomics: Real-World Datasets (ONT, UHGG)

Datasets: (i) GIAB HG002 long-read ONT data (high-error, third-generation); (ii) Unified Human Gut Genome (UHGG) collection (large-scale, low-error NGS).

Results: QA-BPE-seq consistently outperforms baselines across both regimes. ONT (high-error) results:

Table 15. ONT (GIAB HG002) results. Means with 95% confidence intervals over $n = 10$ runs.

Method	Variant F1	Taxa Acc. F1	Recon. Loss	Inf. Time (ms/seq)
Standard BPE	0.795 ± 0.006	0.812 ± 0.007	0.388 ± 0.012	11.5 ± 0.3
SentencePiece	0.801 ± 0.005	0.825 ± 0.006	0.371 ± 0.011	11.6 ± 0.4
WordPiece	0.798 ± 0.006	0.819 ± 0.007	0.379 ± 0.013	11.5 ± 0.3
DNABERT-k (6-mer)	0.823 ± 0.004	0.846 ± 0.005	0.352 ± 0.010	11.2 ± 0.3
QA-BPE-seq (100%)	$\mathbf{0.864 \pm 0.005}$	$\mathbf{0.881 \pm 0.004}$	$\mathbf{0.305 \pm 0.009}$	$\mathbf{11.8 \pm 0.4}$
QA-BPE-seq (70%)	0.830 ± 0.005	0.845 ± 0.004	0.345 ± 0.009	11.9 ± 0.4
QA-BPE-seq (50%)	0.795 ± 0.006	0.810 ± 0.005	0.380 ± 0.010	12.0 ± 0.4
QA-BPE-seq (30%)	0.750 ± 0.006	0.760 ± 0.005	0.420 ± 0.011	12.1 ± 0.5

NGS (UHGG) results:

G.8. Finance: High-Frequency Equities (AAPL)

Dataset and Setup: High-frequency LOB data for AAPL from LOBSTER.

Results: QAT-QF scales to equities, improving predictive and trading metrics over baselines.

G.9. Finance: Rolling-Window Temporal Robustness (BTC/USD, Full Year 2023)

To demonstrate temporal robustness beyond a single quarter, we extend our BTC/USD evaluation across all four quarters of 2023 using a strict rolling-window protocol. For each quarter, the vocabulary and downstream models are trained only on data preceding that quarter.

Table 16. UHGG (NGS) results. Means with 95% confidence intervals over $n = 10$ runs.

Method	Variant F1	Taxa Acc. F1	Recon. Loss	Inf. Time (ms/seq)
Standard BPE	0.852 ± 0.003	0.881 ± 0.004	0.295 ± 0.008	9.8 ± 0.2
SentencePiece	0.860 ± 0.003	0.893 ± 0.004	0.280 ± 0.007	9.9 ± 0.2
WordPiece	0.855 ± 0.004	0.887 ± 0.005	0.286 ± 0.009	9.8 ± 0.3
DNABERT-k (6-mer)	0.875 ± 0.002	0.908 ± 0.003	0.264 ± 0.006	9.5 ± 0.2
QA-BPE-seq (100%)	$\mathbf{0.915 \pm 0.003}$	$\mathbf{0.935 \pm 0.003}$	$\mathbf{0.221 \pm 0.005}$	$\mathbf{10.1 \pm 0.3}$
QA-BPE-seq (70%)	0.878 ± 0.004	0.898 ± 0.004	0.250 ± 0.007	10.2 ± 0.3
QA-BPE-seq (50%)	0.842 ± 0.005	0.860 ± 0.005	0.276 ± 0.008	10.3 ± 0.3
QA-BPE-seq (30%)	0.790 ± 0.006	0.805 ± 0.006	0.310 ± 0.009	10.5 ± 0.4

Table 17. AAPL high-frequency results. Means with 95% confidence intervals over $n = 10$ runs.

Method	Ret. Pred. (%)	Vol. RMSE	Regime Acc. (%)	Sharpe	Inf. Time (ms/seq)
Standard BPE	63.1 ± 0.6	0.0125 ± 0.0004	75.8 ± 0.7	1.41 ± 0.06	14.8 ± 0.4
SAX	61.5 ± 0.7	0.0121 ± 0.0005	77.0 ± 0.6	1.38 ± 0.07	14.2 ± 0.3
BOSS	64.0 ± 0.5	0.0113 ± 0.0004	80.1 ± 0.5	1.53 ± 0.06	14.5 ± 0.4
QAT-QF	$\mathbf{69.8 \pm 0.5}$	$\mathbf{0.0085 \pm 0.0003}$	$\mathbf{87.9 \pm 0.4}$	$\mathbf{1.81 \pm 0.08}$	$\mathbf{15.0 \pm 0.5}$

Table 18. Rolling-window out-of-sample Sharpe ratios for BTC/USD across 2023. Each quarter uses models trained strictly on preceding data. Means with 95% confidence intervals over $n = 10$ runs.

Quarter	QAT-QF Sharpe	BPE Sharpe	Δ (%)	Market Context
Q1 2023	1.72 ± 0.07	1.32 ± 0.05	+30.3	Recovery phase
Q2 2023	1.58 ± 0.09	1.21 ± 0.06	+30.6	Consolidation
Q3 2023	1.45 ± 0.08	1.15 ± 0.07	+26.1	High volatility
Q4 2023	1.68 ± 0.10	1.29 ± 0.06	+30.2	Bull market
Average	**1.61**	**1.24**	**+29.8**	—

Key Observations: (i) QAT-QF maintains consistent improvements (+26–31%) across all market regimes. (ii) Q3 2023 exhibited elevated volatility (VIX-equivalent spike); QAT-QF gains persist (+26.1%), demonstrating cross-regime robustness. (iii) The consistency across four quarters with varying market conditions validates generalization beyond a single test period.

H. Domain-Specific Instantiations

We now detail the instantiation of the QA-Token framework for three distinct domains: genomic sequencing, social media text, and quantitative finance. Detailed pseudocode algorithms for each domain are provided in Section H.9.

H.1. Genomics (QA-BPE-seq)

Context: This instantiation targets the analysis of DNA or RNA sequencing reads, which are often affected by base-calling errors, for applications such as genetic variant calling, taxonomic classification, or sequence modeling. **Atomic Elements & Quality:** The base alphabet is $\Sigma = \{\text{A, C, G, T/U, N}\}$. The primary quality information for each atomic base s_i comes from Phred scores $Q_{\text{phred},i}$. The error probability is $P_{\text{error}}(i) = 10^{-Q_{\text{phred},i}/10}$, leading to an atomic quality score $q_i = 1 - P_{\text{error}}(i)$. To model read end quality degradation, for a base at position i (0-indexed) in a read of length L, the position-adjusted quality is:

$$q'_i = q_i \cdot \exp\left(-\beta_{\text{pos}} \cdot \frac{|i - (L-1)/2|}{(L-1)/2 + \epsilon_{len}}\right) \tag{41}$$

where $\beta_{\text{pos}} \geq 0$ is a learnable parameter in θ_{adapt}. **Token Quality (q_t):** For a token $t = s_1 ... s_{|t|}$, we use the geometric mean of the position-adjusted atomic qualities to compute its aggregated scalar quality: $q_t = (\prod_{j=1}^{|t|} q'_{s_j})^{1/|t|}$. The geometric mean is sensitive to low-quality bases. This q_t is used for the constituent qualities q_a and q_b in the merge score (Eq. 5). **Merge Score (w_{ab}):** The score is calculated using Equation 5, with the geometric mean qualities q_a, q_b, the learnable parameter $\alpha \in \theta_{\text{adapt}}$, and $\psi(a, b) = 1$. **Reward Components (R_{genomic}):** The overall reward (Eq. 26) uses weights $\lambda_j \in \theta_{\text{adapt}}$. Specific raw components R^{raw} include:

- $R_Q^{\text{raw}}(a, b)$: Quality of the newly formed token t_{ab}. This is its geometric mean quality: $R_Q^{\text{raw}}(a, b) = q_{ab} = (\prod_{l=1}^{|a|+|b|} q'_{s_{ab,l}})^{1/(|a|+|b|)}$.
- $R_I^{\text{raw}}(a, b)$: Log-ratio of probabilities: $R_I^{\text{raw}}(a, b) = \log \frac{P(t_{ab})}{P(a)P(b)+\epsilon_p}$.
- $R_C^{\text{raw}}(a, b)$: Complexity penalty: $R_C^{\text{raw}}(a, b) = -|t_{ab}|$.
- R_{bio}^{raw} (Optional): A domain-specific reward based on overlap with known genomic features (e.g., genes, regulatory elements from databases like dbSNP (Sherry et al., 2001)).

Raw components are normalized using the adaptive EMA method (Eq. 35). **Adaptive Parameters (θ_{adapt}):** Includes α, β_{pos}, reward weights λ_j, and potentially parameters for soft frequency/quality gating. **Algorithm:** The two-stage learning process (Section E) is applied. An RL policy is optimized (Algorithm 1), and then the adaptive parameters θ_{adapt} are learned (Algorithm 2) by optimizing a downstream task objective.

H.2. Quantitative Finance (QAT-QF)

Context: This instantiation focuses on analyzing noisy, non-stationary high-frequency financial data for tasks like forecasting price movements or developing trading strategies. **Atomic Elements & Quality:** Atomic elements s_i are discretized events from high-frequency data (e.g., fixed-length segments of LOB events). Each atomic element s_i is assigned a scalar quality score $q_i = \sum_k w_k q_{k,i}$, where $q_{k,i}$ are normalized quality components (e.g., $q_{\text{snr}}, q_{\text{liq}}$) and w_k are learnable weights in θ_{adapt}. **Token Quality (q_t):** For a token t composed of atomic elements $\{s_i\}_{i \in t}$, the aggregated scalar quality is the arithmetic mean: $q_t = \frac{1}{|t|} \sum_{i \in t} q_i$. This is used for q_a, q_b in the merge score. **Merge Score (w_{ab}):** Calculated using Equation 5, with q_a, q_b, learnable $\alpha \in \theta_{\text{adapt}}$, and $\psi(a, b) = 1$. **Market Regimes:** An identified regime indicator can condition the RL policy and reward components. **Reward Components (R_{finance}):** Raw components R^{raw} are normalized using the adaptive EMA method.

- $R_Q^{\text{raw}}(a, b)$: Length-weighted average quality: $R_Q^{\text{raw}}(a, b) = \frac{|a|q_a + |b|q_b}{|a|+|b|}$.

- $R_I^{\text{raw}}(a,b)$: Information reward blended across regimes: $R_I^{\text{raw}}(a,b) = \gamma_{\text{regime}} \cdot I_{\text{normal}}(a,b) + (1-\gamma_{\text{regime}}) \cdot I_{\text{stress}}(a,b)$, where $I_{\text{regime}} = \log \frac{P(t_{ab}|\text{regime})}{P(a|\text{regime})P(b|\text{regime})+\epsilon_p}$. The blending factor γ_{regime} is a learnable parameter in θ_{adapt}.
- $R_P^{\text{raw}}(a,b)$: Predictive Power (Mutual Information with future returns):

$$R_P^{\text{raw}}(a,b) = \frac{\text{MI}(t_{ab}, \text{Disc}(R_\tau))}{\text{NormFactor}_{MI} + \epsilon_{MI}} \tag{42}$$

  $\text{Disc}(R_\tau)$ is discretized future return. NormFactor_{MI} is an adaptive normalization factor.
- $R_C^{\text{raw}}(a,b)$: Complexity penalty with volatility scaling:

$$R_C^{\text{raw}}(a,b) = -\left(|t_{ab}| \cdot \log(|V_k|+1) \cdot \text{VolScale}\right) \tag{43}$$

  where VolScale depends on a learnable parameter $\beta_{\text{vol}} \in \theta_{\text{adapt}}$.

Adaptive Parameters (θ_{adapt}): Includes α, quality component weights w_k, β_{vol}, γ_{regime}, and reward weights λ_j. **Algorithm:** The two-stage learning process is applied as in the genomics domain.

H.3. Social Media Text (QA-BPE-nlp)

Context: This instantiation addresses the challenges of processing noisy user-generated text for tasks such as sentiment analysis or NER. **Atomic Elements & Quality:** The base alphabet consists of characters. Quality for a token t is modeled using a multi-dimensional vector $\boldsymbol{q}_t = (q_{\text{orth}}(t), q_{\text{sem}}(t), \dots)$ detailed in Appendix D.3. The aggregated scalar quality is $q_t = \sum_j w_j \boldsymbol{q}_{t,j}$, where $w_j \geq 0$ are learnable weights in θ_{adapt}. **Token Quality (q_t):** The aggregated score q_t is used for q_a, q_b in the merge score. **Merge Score (w_{ab}):** Calculated using Equation 5 with q_a, q_b, learnable $\alpha \in \theta_{\text{adapt}}$, and a semantic compatibility factor $\psi(a,b)$:

$$\psi(a,b) = \exp(\beta_{sem} \cdot \text{cosine}(\boldsymbol{v}_a, \boldsymbol{v}_b)) \tag{44}$$

where $\boldsymbol{v}_a, \boldsymbol{v}_b$ are pre-trained embeddings and $\beta_{sem} \geq 0$ is a learnable parameter in θ_{adapt}. **Noise Models:** Probabilistic models $P(t'|t)$ capturing likely variations inform the noise robustness reward R_N. **Reward Components (R_{social}):** Raw components are normalized before being weighted by λ_j.

- $R_Q^{\text{raw}}(a,b)$: Blend of compositional and direct quality: $R_Q^{\text{raw}}(a,b) = \omega \frac{|a|q_a+|b|q_b}{|a|+|b|} + (1-\omega)q_{ab}$, with learnable blending weight $\omega \in [0,1]$.
- $R_S^{\text{raw}}(a,b)$: Semantic Coherence: $\text{PMI}(a,b) \cdot \text{cosine\_similarity}(\boldsymbol{v}_a, \boldsymbol{v}_b)$.
- $R_N^{\text{raw}}(a,b)$: Noise Robustness: $R_{\text{noise}}(t_{ab}) - \frac{|a|R_{\text{noise}}(a)+|b|R_{\text{noise}}(b)}{|a|+|b|}$, based on the noise model.
- $R_C^{\text{raw}}(a,b)$: Complexity penalty: $R_C^{\text{raw}}(a,b) = -|t_{ab}|$.
- $R_V^{\text{raw}}(a,b)$: Vocabulary Efficiency: $\frac{\log(1+f(t_{ab}))}{|t_{ab}|}$.

Adaptive Parameters (θ_{adapt}): Includes α, β_{sem}, quality dimension weights w_j, reward weights λ_j, and the blending weight ω. **Algorithm:** The two-stage learning process is applied as in the other domains.

H.4. Financial Experimental Methodology Details

All trading simulations and return prediction evaluations for the quantitative finance domain (Section 5.2) were conducted with rigorous attention to backtesting best practices to ensure the validity of results and avoid common pitfalls.

- **Walk-Forward Validation:** A strict walk-forward validation scheme was employed. The dataset was divided into chronological segments. For each segment k, the model (including the QA-Token vocabulary construction and downstream predictive/trading model) was trained on data up to the start of segment k, validated on segment $k-1$ (or a dedicated validation portion of the training data), and then tested out-of-sample only on segment k. The training window was then rolled forward to include segment k for training before testing on segment $k+1$. This process ensures that the model is always tested on data not seen during its training or hyperparameter tuning phases for that specific test period.

- **Lookahead Bias Prevention:** Extreme care was taken to prevent any form of lookahead bias. All features, quality scores, token definitions, and trading decisions at any time t were based strictly on information available up to and including time $t-1$. Future return labels ($R_{t+\tau}$) used for training predictive models or as part of the R_P reward component were sourced from periods strictly after the information used for input features and token construction.
- **Test Set and Data Splitting:** The overall dataset (BTC/USD LOB data, Q1 2023) was split chronologically: 70% for the initial training pool, 15% for validation (used for hyperparameter tuning of downstream models and early stopping), and the final 15% (approximately 2 weeks of 1-minute data) as the ultimate out-of-sample test set for reporting final performance metrics like Sharpe Ratio and prediction accuracy. This test set was held out and used only once after all model development and tuning.
- **Transaction Costs:** A realistic transaction cost of 5 basis points (0.05%) per trade was applied to simulate market friction. This cost was deducted for both buying and selling actions in the trading simulations.
- **PPO Trading Agent Details:** The PPO-based trading agent used a 2-layer MLP policy network and a separate 2-layer MLP value network, each with 128 hidden units and ReLU activation functions. The input to these networks consisted of a sequence of recent token embeddings (generated by QAT-QF or baseline tokenizers from the LOB data) and the agent's current market position (long, short, or flat). The agent's action space was discrete (buy, sell, hold). The reward function for the PPO agent was the realized profit and loss (PnL) from its trades over a short horizon, adjusted for transaction costs. Standard PPO hyperparameters were used, including a clipping parameter $\epsilon = 0.2$, GAE $\lambda = 0.95$, and an entropy bonus for exploration. The PPO agent was re-trained periodically within the walk-forward scheme.
- **Details for R_P^{raw} Reward (Eq. 42):** The parameter M_{MI} (window for NormFactor$_{MI}$) was set to 1000 merge steps in our experiments. The future return R_τ was for $\tau = 5$ minutes ahead and discretized into 3 bins (negative, neutral, positive) based on empirical quantiles from the training data.

H.5. Detailed Reward Components

The general structure of the reward $R(a,b)$ for merging tokens a and b into $t_{merged} = a||b$ is: $R(a,b) = \sum_j \lambda_j \hat{R}_j(a,b)$, where $\hat{R}_j$ are adaptively normalized components (see Section 4.2). The weights $\lambda_j \geq 0$ (parameterized via $\boldsymbol{\beta}_{\lambda_j}$ and softmax) are part of θ_{adapt}.

H.6. Common Components

- $R_Q^{\text{raw}}(a,b)$: Raw Quality reward. This component incentivizes merges that result in high-quality tokens. A common formulation for the raw component is the length-weighted arithmetic mean of the qualities of the constituent tokens a and b:

$$R_Q^{\text{raw}}(a,b) = \frac{|a|q_a + |b|q_b}{|a| + |b|} \tag{45}$$

  where q_a, q_b are the quality scores of tokens a, b respectively, and $|a|, |b|$ are their lengths. For Social Media, a blended approach might be used for $R_Q^{\text{raw}}(a,b)$:

$$R_Q^{\text{raw}}(a,b) = \omega \left(\frac{|a|Q_{agg}(a) + |b|Q_{agg}(b)}{|a| + |b|} \right) + (1-\omega) Q_{agg}(a||b) \tag{46}$$

  where $Q_{agg}(t)$ is the aggregate quality score for token t (from Section D.3) and $\omega \in [0,1]$ is a learnable blending weight in θ_{adapt}.

- $R_I^{\text{raw}}(a,b)$: Raw Information gain. This rewards merges that are statistically significant. A common formulation:

$$R_I^{\text{raw}}(a,b) = \log \frac{f(t_{merged})}{f(a)f(b) + \epsilon_f} \tag{47}$$

  where $f(\cdot)$ denotes frequency and $\epsilon_f > 0$ (e.g., 10^{-8}) is for stability. For Finance, this can be blended based on market regime: $R_I^{\text{raw}}(a,b) = \gamma_{\text{regime}} I_{\text{normal}} + (1-\gamma_{\text{regime}}) I_{\text{stress}}$, where $I_{\text{regime}} = \log \frac{f(t_{merged}|M=\text{regime})}{f(a|M=\text{regime})f(b|M=\text{regime})+\epsilon_f}$. $\gamma_{\text{regime}} \in [0,1]$ is a learnable parameter in θ_{adapt}.

- $R_C^{\text{raw}}(a,b)$: Raw Complexity penalty. This penalizes overly complex vocabularies and is typically negative. A common formulation:
$$R_C^{\text{raw}}(a,b) = -\text{len}(t_{merged}) \cdot \log(|V_t|+1) \cdot [\text{ScalingFactor}] \tag{48}$$
For Finance, the ScalingFactor can incorporate market volatility using $\beta_{vol} \in \theta_{adapt}$ as per Equation 43.

H.7. Domain-Specific Components

- **Genomics:** $R_{bio}^{\text{raw}}(a,b) = \text{Score}_{\text{Overlap}}(t_{merged}, \text{KnownBiologicalFeatures})$. A positive reward if t_{merged} significantly overlaps with known biological features (e.g., genes from GENCODE (Harrow et al., 2012), variants from dbSNP (Sherry et al., 2001)). The overlap score was calculated as the Jaccard index between the character span of the merged token t_{merged} and the character span of known genomic features. A higher Jaccard index, indicating greater overlap, results in a higher reward.
- **Finance:**
  - $R_P^{\text{raw}}(a,b)$: Predictive Power:
$$R_P^{\text{raw}}(a,b) = \frac{\text{MI}(t_{\text{merged}}; \text{Disc}(R_\tau))}{\text{NormFactor}_{MI} + \epsilon_{MI}} \tag{49}$$
Uses Mutual Information (MI) $\text{MI}(X;Y) = \sum_{x\in X, y\in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$. R_τ is the discretized future return (e.g., 3 bins for $\tau = 5$ min based on empirical quantiles from the training data). NormFactor_{MI} is the adaptively calculated 95th percentile of MI values from candidate pairs over the last M_{MI} (e.g., 1000) merge steps within the current RL episode. $\epsilon_{MI} > 0$ (e.g., 10^{-8}). While this adaptive normalization of MI introduces a degree of non-stationarity to the R_P reward component within an RL episode, it was found that standard PPO training handled this adequately. The responsiveness of the reward to the informativeness of newly forming tokens was deemed beneficial, and the M_{MI} window provides some smoothing. Alternatives using a fixed normalization factor (e.g., derived from an initial global scan of MI values) were found to be less responsive to the changing characteristics of tokens as the vocabulary evolved during the RL episode.
- **Social Media:**
  - $R_S^{\text{raw}}(a,b)$: Semantic Coherence: $\text{PMI}(a,b) \cdot \text{cosine\_similarity}(\boldsymbol{v}_a, \boldsymbol{v}_b)$. Pre-trained embeddings $\boldsymbol{v}_a, \boldsymbol{v}_b$ (e.g., fastText (Bojanowski et al., 2017)).
  - $R_N^{\text{raw}}(a,b)$: Noise Robustness:
$$\left(R_{\text{noise}}(t_{\text{merged}}) - \frac{|a|R_{\text{noise}}(a) + |b|R_{\text{noise}}(b)}{|a|+|b|} \right), \tag{50}$$
where $R_{noise}(t) = 1 - \mathbb{E}_{t'\sim P(\cdot|t)}[\text{normalized\_edit\_distance}(t,t')]$ based on noise model $P(t'|t)$ (Appendix H.8).
  - $R_V^{\text{raw}}(a,b)$: Vocabulary Efficiency: $\frac{\log(1+f(t_{\text{merged}}))}{|t_{\text{merged}}|}$.

H.8. Further Details on Social Media Noise Models

Formalizing linguistic noise for social media text involves defining probabilistic transformations $P(t'|t)$ from a canonical form t to an observed variant t' (Han et al., 2013). These models inform the noise robustness measure $R_{\text{noise}}(t)$ (defined in Appendix H.5, Eq. 50). $P(t'|t)$ was constructed based on heuristic rules derived from commonly observed error patterns in social media text and principles outlined in existing literature on noisy text processing. The specific noise types modeled include:

- **Character-Level Noise:**
  - **Repetition:** Probability of a character c being realized as c^n (a sequence of n identical characters). For $n \geq 1$, this can be modeled using a geometric-like distribution. If p_{stop} is the probability of not repeating an additional time: $P(c \to c^n) = (1-p_{stop})^{n-1} \cdot p_{stop}$. The parameter p_{stop} was set empirically to 0.5, allowing for moderate repetitions common in social media (e.g., "soooo goood").

  - **Substitution:** $P(c_i \to c_j) = M_{\text{sub}}[c_i, c_j]$, where M_{sub} is a confusion matrix. M_{sub} was constructed heuristically, assigning higher probabilities to substitutions between characters that are adjacent on a standard QWERTY keyboard layout and to common phonetic misspellings (e.g., 'c' vs 'k'). Off-diagonal probabilities were generally small.
  - **Omission (Deletion):** $P(c \to \epsilon) = p_{\text{del}}(c)$ is the character-specific deletion probability. This was set to a small uniform value (e.g., $p_{\text{del}}(c) = 0.01$) for all characters, reflecting occasional accidental omissions.

- **Word-Level Noise:**
  - **Abbreviation:** $P(w \to \text{abbr}(w)) = f_{\text{abbr}}(w \to \text{abbr}(w))$. This probability was derived from a compiled dictionary of common internet slang and abbreviations sourced from publicly available online linguistic resources. For words in this dictionary, f_{abbr} was set to a moderate value (e.g., 0.3), and zero otherwise.
  - **Phonetic Substitution:** $P(w_1 \to w_2) \propto \exp(\lambda_{\text{phon}} \cdot \text{phon\_sim}(w_1, w_2))$. The phonetic similarity $\text{phon\_sim}(w_1, w_2)$ was computed using the Double Metaphone algorithm. The scaling factor λ_{phon} was set to 1.0.

- **Discourse-Level Noise (examples):** For the experiments reported in this paper, the noise modeling primarily focused on character-level and word-level phenomena, as these are highly prevalent and tractable to model. Explicit modeling of discourse-level noise, such as code-switching or complex punctuation patterns, was considered beyond the scope of the current noise component R_N, though it represents an interesting avenue for future work.

These probabilistic models are used to define $P(t'|t)$, which is then used to compute the expected distance in the noise robustness measure $R_{\text{noise}}(t) = 1 - \mathbb{E}_{t' \sim P(\cdot|t)}[\text{dist}_{\text{norm}}(t, t')]$. The normalized distance metric $\text{dist}_{\text{norm}}(t, t')$ used was the Levenshtein distance divided by the maximum length of the two strings t and t'.

H.9. Domain-Specific Algorithms

This section provides detailed pseudocode for the QA-Token framework as instantiated for Quantitative Finance, Genomics, and Social Media. These algorithms complement the domain instantiations described in Section H, illustrating the core mechanics within each domain.

H.9.1. QUANTITATIVE FINANCE (QAT-QF)

Algorithm 7 Quality-Aware Tokenization Merge Score and Reward Calculation (QAT-TOKEN - Finance)

Require: Current vocabulary V_t, corpus statistics (frequencies $f(\cdot)$), current adaptive parameters $\theta_{adapt} = \{\alpha, \beta_{vol}, \gamma_{\text{regime}}, f_{min}, \delta_{\text{gate}}, w_k \text{ (param by } \boldsymbol{\beta}_w)\}$, reward weights $\lambda_Q, \lambda_I, \lambda_P, \lambda_C$.
Ensure: For each candidate merge pair (a, b): quality-aware merge score w_{ab}, total immediate reward $R(a, b)$.
Identify candidate merge pairs C_t from corpus (e.g., from priority queue PQ_t).
for each adjacent token pair $(a, b) \in C_t$ **do**
  Let $t_{merged} \leftarrow a||b$.
  Retrieve/compute frequencies $f(a)$, $f(b)$, and $f(a, b)$.
  Retrieve/compute average qualities q_a, q_b (using $Q[i]$ from Section D.2, aggregated for tokens a, b, and weights $w_k = \text{softmax}(\boldsymbol{\beta}_w)_k$).
  Quality-Aware Merge Score (w_{ab}): $w_{ab} \leftarrow \frac{f(a,b)}{f(a)\cdot f(b)+\epsilon_f} \cdot \left(\left(\frac{q_a+q_b}{2} + \epsilon_Q\right)^{\alpha}\right) \cdot \psi(a, b)$ // $\psi(a, b) = 1$ for finance
  Frequency Gating (Optional): // Frequency gating not used in final experiments $\tilde{f}(a, b) \leftarrow f(a, b)$.
  $R_Q^{\text{raw}}(a, b) \leftarrow \frac{|a|\cdot q_a+|b|\cdot q_b}{|a|+|b|}$.
  Estimate I_{normal}, I_{stress} based on regime-conditioned $\tilde{f}(a, b)$. $R_I^{\text{raw}}(a, b) \leftarrow \gamma_{\text{regime}} \cdot I_{normal} + (1 - \gamma_{\text{regime}}) \cdot I_{stress}$.
  $MI_{val} \leftarrow \text{MI}(t_{merged}; \text{Disc}(R_\tau))$. $R_P^{\text{raw}}(a, b) \leftarrow \frac{MI_{val}}{\text{NormFactor}_{MI}+\epsilon_{MI}}$ (NormFactor$_{MI}$ from Section H.2).
  $\sigma_{curr}, \sigma_{hist} \leftarrow \text{GetVolatility}()$; $VolScaling \leftarrow (1 + \max(0, (\sigma_{curr} - \sigma_{hist})/(\sigma_{hist} + \epsilon_{\text{vol}})))^{\beta_{vol}}$
  $R_C^{\text{raw}}(a, b) \leftarrow -|t_{merged}| \cdot \log(|V_t| + 1) \cdot VolScaling$
  Normalize raw rewards: $\hat{R}_j(a, b) \leftarrow \text{AdaptiveNormalize}(R_j^{\text{raw}}(a, b))$ using Eqs. 35, 33, and 34.
  Total Immediate Reward ($R(a, b)$): $R(a, b) \leftarrow \sum_j \lambda_j \hat{R}_j(a, b)$.
  Store w_{ab}, $R(a, b)$, and other features for (a, b) for policy input or selection.
end for

Algorithm 8 Adaptive Parameter Learning for QA-TOKEN (Finance)

Require: Training dataset $\mathcal{D}_{\text{train}}$; Downstream task loss function $L_{\text{task}}(\cdot, \cdot)$; Model params Θ_{model}; Initial adaptive parameters θ_{adapt}; Learning rate η_θ; Epochs E_{adapt}; Gumbel-Softmax τ_g.
Ensure: Optimized adaptive parameters θ^*_{adapt}.
Initialize θ_{adapt}.
for each adaptation epoch $e = 1, \ldots, E_{adapt}$ **do**
  for each mini-batch $B = \{(S_{\text{seq},i}, Y_{\text{target},i})\}$ from $\mathcal{D}_{\text{train}}$ **do**
    $\mathcal{S}'_{batch} \leftarrow \text{SOFTTOKENIZEGUMBEL}(B, \theta_{adapt}, \tau_g)$ // Eq. 37
    $L_{\text{batch\_task}} \leftarrow L_{\text{task}}(\mathcal{S}'_{batch}, \{Y_{\text{target},i}\}, \Theta_{\text{model}})$
    if regularization $L_{\text{reg}}(\theta_{adapt})$ is used **then**
      $L_{\text{total\_batch}} \leftarrow L_{\text{batch\_task}} + L_{\text{reg}}(\theta_{adapt})$
    else
      $L_{\text{total\_batch}} \leftarrow L_{\text{batch\_task}}$
    end if
    Compute gradients $\nabla_{\theta_{adapt}} L_{\text{total\_batch}}$. // Uses Gumbel-Softmax trick
    Update $\theta_{adapt} \leftarrow \theta_{adapt} - \eta_\theta \nabla_{\theta_{adapt}} L_{\text{total\_batch}}$.
    Apply constraints to θ_{adapt} (e.g. $\alpha \geq 0$, softmax for weights).
  end for
  Anneal τ_g.
end for

return $\theta^*_{adapt} \leftarrow \theta_{adapt}$.

H.9.2. GENOMICS (QA-BPE-SEQ)

Algorithm 9 Reward Calculation for a Merge (Genomics)

Require: Tokens a, b with qualities q_a, q_b; frequencies $f(\cdot)$; reward weights λ_j from θ_{adapt}. For genomics, q_a, q_b represent geometric mean qualities of constituent tokens.
Ensure: Raw rewards $R_j^{\text{raw}}(a, b)$ for merging a and b.
$t_{merged} \leftarrow a||b$
$R_Q^{\text{raw}}(a, b) \leftarrow (\prod_{l=1}^{|t_{merged}|} q'_{s_{merged,l}})^{1/|t_{merged}|}$. // Geometric mean quality
$R_I^{\text{raw}}(a, b) \leftarrow \log \frac{f(t_{merged})}{f(a) \cdot f(b) + \epsilon_f}$.
$R_C^{\text{raw}}(a, b) \leftarrow -\text{len}(t_{merged})$.
if Biological Reward is used **then**
  $OverlapScore \leftarrow \text{ComputeOverlapScore}(t_{merged}, \text{KnownBiologicalFeatures})$.
  $R_{bio}^{\text{raw}}(a, b) \leftarrow OverlapScore$.
end if

return All relevant $R_j^{\text{raw}}(a, b)$. (Normalized rewards $\hat{R}_j$ computed later using Eq. 35).

The size of the RL agent's action space, K_{PQ} (the number of top pairs from the priority queue considered at each step), was set to $K_{PQ} = 50$. This value was chosen based on preliminary experiments indicating it offered a good trade-off between exposing the RL agent to a diverse set of high-potential merges and maintaining a manageable action space size for efficient policy learning. Values explored in the range $[20, 100]$ showed that performance was relatively robust for $K_{PQ} \in [40, 60]$, with smaller values risking premature pruning of potentially beneficial long-term merges and larger values not yielding significant gains while increasing computational cost per policy step. The chosen value of 50 balanced these considerations effectively across domains.

- **RL (PPO specifics) - Stage 1:**
  - Policy/Value MLP Architecture: 2-3 hidden layers, each with 128-512 units. Activation functions: ReLU or Tanh.
  - PPO ϵ_{clip} (clipping parameter): $[0.1, 0.3]$, typically 0.2.
  - GAE λ_{GAE} (Generalized Advantage Estimation lambda): $[0.9, 0.99]$, typically 0.95.
  - Discount factor γ_{RL}: $[0.95, 1.0]$, often 0.99 for non-terminating tasks or long horizons.
  - Optimizer: Adam (Kingma & Ba, 2014). Learning rates η_π (policy), η_v (value): $[1 \times 10^{-5}, 5 \times 10^{-4}]$.
  - Entropy bonus coefficient c_S (or c_2): $[0.0, 0.05]$, typically 0.01.
  - Value function loss coefficient c_{VF} (or c_1): $[0.25, 1.0]$, typically 0.5.
  - Batch size (number of transitions per update): $[128, 4096]$ or more, depending on data/memory.
  - PPO epochs per update (passes over collected data): $[3, 20]$, typically $4 - 10$.
  - Number of actors / parallel environments: 1 to N_{cores} or N_{GPUs}.
- **Adaptive Reward Normalization (Section 4.2):**
  - EMA momentum β_{norm}: $[10^{-3}, 10^{-1}]$, typically 10^{-2}.
  - ϵ_R (stability constant): Typically 10^{-8}.
- **Reward Weights ($\boldsymbol{\beta}_{\lambda_j}$ leading to λ_j):** Initial values for $\boldsymbol{\beta}_{\lambda_j}$ in $\theta_{\text{adapt}}^{(0)}$ for Stage 1 can be zero or small random numbers (resulting in uniform or near-uniform λ_j). These are then optimized in Stage 2.
- **Adaptive Learning Parameters (θ_{adapt} from Algorithm 2) - Stage 2:**
  - Optimizer: Adam. Learning rate $\eta_\theta \in [1 \times 10^{-6}, 1 \times 10^{-4}]$.
  - Gumbel-Softmax temperature τ: Annealed from an initial high value (e.g., $1.0 - 5.0$) down to a small positive value (e.g., $0.1 - 0.5$) over training. Schedule: e.g., exponential decay $\tau_t = \max(\tau_{final}, \tau_0 \cdot d^t)$.
  - Logit composite function (Eq. 37): Norm_ℓ is typically identity or batch normalization if logits vary widely.

- **Domain-Specific Adaptive Parameters and Quality Metric Settings:**
  - **Genomics Specific:**
    - β_{pos} (positional quality decay): Learned. Initial range explored $[0.001, 0.1]$.
    - ϵ_{len} (Eq. 41): 10^{-6}.
  - **Social Media Specific:**
    - $\boldsymbol{\beta}_{w_j}$ (for Q_{agg} weights w_j): Learned.
    - β_{sem} (semantic compatibility, Eq. 44): Learned. Initial range $[0.1, 5.0]$.
    - ω (blending weight for R_Q^{raw}, Eq. 46): Learned. Parameterized via sigmoid of an unconstrained variable.
    - Note: The direct downstream loss component R_D was not used in the RL reward for the final reported Social Media NLP experiments (Section H.3).
  - **Finance Specific:**
    - $\boldsymbol{\beta}_{w_k}$ (for $Q[i]$ weights w_k): Learned.
    - β_{vol} (volatility scaling in R_C): Learned. Initial range $[0.0, 2.0]$.
    - γ_{regime} (regime blending for R_I): Learned. Parameterized via sigmoid of an unconstrained variable.
    - M_{MI} (window for NormFactor$_{MI}$): e.g., 1000 steps.
    - Note: Soft frequency gating was disabled in the final configuration for Quantitative Finance experiments (Section 5.2).
- **General QA-Token Parameters:**
  - ϵ_f, ϵ_Q (Eq. 5): 10^{-8}.
  - α (quality sensitivity in w_{ab}): Learned. Initial range $[0.0, 5.0]$.
- **Vocabulary Settings:**
  - Target vocabulary size V_{target}: Typically $[16000, 64000]$.

H.9.3. Converged Adaptive Parameters

Table 19 provides mean converged values ($\pm$ standard deviation over three experimental runs) for key adaptive parameters in θ_{adapt} for each domain. The adaptive learning process tunes these parameters to optimize downstream task performance, leading to domain-specific configurations.

Table 19. Converged Adaptive Parameters ($\pm$ Std Dev).

Parameter	Genomics	Finance	Social Media
α (Quality Sensitivity)	0.72 ± 0.03	0.95 ± 0.03	1.15 ± 0.05
λ_Q (Quality Reward Weight)	0.35 ± 0.03	0.30 ± 0.02	0.33 ± 0.03
λ_I (Information Reward Weight)	0.25 ± 0.02	0.20 ± 0.02	0.22 ± 0.02
λ_C (Complexity Reward Weight)	0.15 ± 0.01	0.10 ± 0.01	0.12 ± 0.01
β_{pos} (Genomics Positional Decay)	0.014 ± 0.002	N/A	N/A
β_{vol} (Finance Volatility Scaling)	N/A	0.50 ± 0.05	N/A
γ_{regime} (Finance Regime Blending)	N/A	0.60 ± 0.04	N/A
w_{orth} (NLP Orthographic Weight)	N/A	N/A	0.32 ± 0.03
w_{sem} (NLP Semantic Weight)	N/A	N/A	0.28 ± 0.02
w_{liq} (Finance Liquidity Weight)	N/A	0.45 ± 0.04	N/A
ω_{social} (NLP Quality Blend)	N/A	N/A	0.55 ± 0.05

H.10. Social Media Ablation Results

Ablation studies in Table 20 are designed to confirm the individual effects of QA-BPE-nlp's quality-aware components. We distinguish the impacts of: (1) the multi-dimensional quality rewards (row 'w/o Quality'), (2) semantic coherence considerations (row 'w/o Semantic'), (3) noise robustness features (row 'w/o Noise'), and (4) adaptive parameter learning (row 'w/o Adaptive Params'). Analysis of the learned weights w_j for the quality dimensions (as detailed with values in Appendix D.3) indicates varying importance across dimensions (e.g., orthogonality q_{orth} and semantics q_{sem} frequently receive higher weights across runs) and reward components λ_i, adapting to the specific task and dataset characteristics.

Table 20. Ablation Study for QA-BPE-nlp on TweetEval Sentiment. Values are means with 95% confidence intervals over $n = 10$ runs.

Configuration	TweetEval Score	Rel. Change (%)
QA-BPE-nlp (Full)	**74.5 ± 0.3**	**-**
w/o RL Framework (Greedy w_{ab})	72.1 ± 0.4	-3.2
w/o Quality ($R_Q = 0$)	71.5 ± 0.5	-4.0
w/o Semantic ($R_S = 0$)	72.8 ± 0.3	-2.3
w/o Noise ($R_N = 0$)	73.2 ± 0.4	-1.7
w/o Vocab Eff ($R_V = 0$)	73.9 ± 0.3	-0.8
w/o Adaptive Params (α, w_j fixed)	71.8 ± 0.5	-3.6
QualTok-nlp (Ablation Baseline)	71.9 ± 0.4	-3.5

I. Dataset, Baseline, and Evaluation Details

This section supplements dataset descriptions, baseline methods, and evaluation metrics discussed in the main paper, providing further details necessary for understanding and reproducing the experimental results reported in Section 5.

I.1. Datasets and Reproducible Evaluation

This subsection details the specific datasets, their versions, and relevant preprocessing steps or configurations used for the experiments reported in Section 5. All datasets are publicly available or available under licenses for academic research.

- **Genomics (QA-BPE-seq Experiments):**
  - **Simulated Human Genomic Reads for Variant Calling, Reconstruction, and Ablations:** Paired-end sequencing reads (150bp) were generated at 30x coverage using the ART simulator (version 2.5.8, using the `art_illumina` tool) (Huang et al., 2012). The simulation was based on the GRCh38 human reference genome (patch 13) and used the built-in HiSeq 2500 error profile (`-ss HS25`). To rigorously assess robustness in high-noise scenarios, as described in Section H.1, the default base error rates (both substitution and indel rates) of this profile were artificially doubled compared to the standard HiSeq 2500 profile. Key ART parameters included: `-p -l 150 -f 30 -m 400 -s 10`. A corpus of approximately 5GB of these synthetic reads was generated and used for training tokenizers, downstream model evaluations, and the ablation studies reported in Section H.1. *Access:* The ART simulator is open-source and available at `https://www.niehs.nih.gov/research/resources/software/art/`. The GRCh38 reference genome can be obtained from public repositories such as NCBI GenBank or Ensembl.
  - **Genome in a Bottle (GIAB) Truth Set for Variant Calling Evaluation:** Variant calling performance was benchmarked against the HG002 truth set (v4.2.1, GRCh38) (Zook et al., 2016). *Access:* GIAB truth sets are publicly available from the NIST FTP site.
  - **CAMI II Metagenome Benchmark for Taxonomic Classification:** Taxonomic classification accuracy was evaluated using the "Toy Human Microbiome Project" (short reads, Assembly Aug2019) dataset from the Second CAMI Challenge (Sczyrba et al., 2017). This benchmark provides datasets with known community compositions and corresponding sequencing reads for performance assessment. *Access:* CAMI II datasets are available through the official CAMI challenge website: `https://data.cami-challenge.org/participate`.
- **Quantitative Finance (QAT-QF Experiments):**
  - **Cryptocurrency Limit Order Book (LOB) Data:** High-frequency Limit Order Book (LOB) data for the BTC/USD trading pair was sourced from LOBSTER (`https://lobsterdata.com/`) (Huang & Polak, 2011), an academic data service. The experiments used reconstructed LOB snapshots at 10 levels for the first quarter of 2023 (Q1 2023). As detailed in Section 5.2, this dataset was split chronologically into 70% for training, 15% for validation, and 15% for out-of-sample testing. Atomic elements for tokenization were defined as sequences of 5 consecutive LOB events, featurized as described in Appendix H.2. *Access:* LOBSTER provides sample data publicly, while full datasets are available under academic or commercial licenses.
- **Social Media Text (QA-BPE-nlp Experiments):**
  - **TweetEval Benchmark:** The TweetEval benchmark (Barbieri et al., 2020) was employed for evaluating QA-BPE-nlp across a diverse set of tweet classification tasks. TweetEval provides a unified framework with standardized data splits (train, validation, test) and evaluation metrics for seven heterogeneous tasks, which are:
    - Emotion Recognition (SemEval-2018 Task 1 (Mohammad et al., 2018))
    - Emoji Prediction (SemEval-2018 Task 2 (Barbieri et al., 2018))
    - Irony Detection (SemEval-2018 Task 3 (Van Hee et al., 2018))
    - Hate Speech Detection (SemEval-2019 Task 5 (Basile et al., 2019))
    - Offensive Language Identification (SemEval-2019 Task 6 (Zampieri et al., 2019))
    - Sentiment Analysis (SemEval-2017 Task 4 (Rosenthal et al., 2017))
    - Stance Detection (SemEval-2016 Task 6 (Mohammad et al., 2016))

    As described in Section I.8, experiments involved fine-tuning a pre-trained BERTweet-base model (Nguyen et al., 2020) on these tasks using different tokenization strategies. *Access:* The TweetEval benchmark, including

data access scripts and details for each constituent dataset, is available on GitHub: https://github.com/cardiffnlp/tweeteval. Access to the underlying tweet content typically requires hydration of tweet IDs and adherence to Twitter's Terms of Service and the respective dataset licenses.

I.2. Dataset and Release Plan

To enable foundation-model training on previously unusable noisy corpora, we will release:

- **Tokenizer artifacts:** Final QA-Token vocabularies, merge tables, and θ_{adapt} for each domain (genomics, finance, social media) at multiple vocabulary sizes.
- **Foundation-model-ready corpora manifests:** Scripts and manifests to reconstruct large noisy pretraining corpora (including filtering and de-duplication), plus sampler configurations matching our 2B-subset tokenizer training protocol.
- **Evaluation suites:** Reproducible pipelines for genomics (variant calling, metagenomics), finance (prediction, volatility, regime, trading), and social media (TweetEval), along with the RL ablation harness.
- **Documentation and governance:** Licenses, data usage considerations, and guidelines for responsible use in high-impact applications (e.g., financial decision-making and clinical genomics).

All code and artifacts will be released under permissive academic licenses to maximize reproducibility and adoption.

I.3. QA-Foundation: Noisy Pretraining Corpora Proposal

We propose QA-Foundation, a curated suite of extremely large, noisy corpora specifically designed to enable foundation-scale pretraining with explicit quality annotations and governance:

- Genomics: multi-petabase metagenomic reads (SRA) with canonicalized metadata, Phred-quality distributions, duplication maps, contamination flags, and per-read provenance hashes. Quality channels include per-base Phred, platform, run, trimming logs, adapter contamination.
- Finance: multi-asset high-frequency LOB streams (equities, futures, crypto) with synchronized calendars, microstructure indicators (spreads, depth, order-imbalance), regime tags, and exchange-specific anomaly flags.
- Social/Web text: multi-platform user-generated text with timestamps, platform labels, de-identified stable author hashes, normalization annotations (hashtags, mentions, URLs), and noise transformations (variant clusters, repetition, keyboard-distance confusion matrices).

Each domain provides standardized schemas, quality channels, and sampling manifests to reproduce tokenizer training at multiple scales (e.g., 0.1%, 1%, 5%) and to support fair comparisons. Scripts produce manifests, deduplication indices (MinHash/LSH), and quality audit reports. Governance includes explicit licenses, intended-use statements, and red-team risk assessments. We will release:

- Tokenizer-ready shards with checksums and integrity manifests
- Quality channel extractors (open-source) and validation suites
- Reproducible samplers that match our 2B-base subset protocol for genomics and analogous budgets for other domains

I.4. Baseline Methods

The following baseline tokenization methods were implemented and configured for rigorous comparison against the proposed QA-Token variants, as presented in Section 5.

- **Standard Byte Pair Encoding (BPE)** (Sennrich et al., 2016): The conventional frequency-based merging algorithm. For genomics and social media experiments, this was implemented using the HuggingFace 'tokenizers' library (version 0.15.0), specifically configured with $tokenizers.models.BPE(unk\_token = "[UNK]", min\_frequency = 2)$, unless stated otherwise. For quantitative finance experiments, a comparable standard BPE implementation was used.

- **SentencePiece** (Kudo & Richardson, 2018): An unsupervised text tokenizer and detokenizer. For genomics and social media experiments, SentencePiece (version 0.1.99) was used in its byte-level BPE mode, operating directly on raw text.
- **WordPiece** (Wu et al., 2016): The subword tokenization algorithm famously used in BERT. It iteratively builds a vocabulary by merging pairs that maximize the likelihood of the training data under a unigram language model assumption.
- **DNABERT k-mer** (Ji et al., 2021): For experiments in the genomics domain, fixed k-mer tokenization was employed as a strong baseline, specifically using 6-mers. This aligns with common practice in models like DNABERT.
- **Symbolic Aggregate approXimation (SAX)** (Lin et al., 2003): A well-established symbolic representation method for time series data, applied in quantitative finance experiments. The mid-price series was discretized using a Piecewise Aggregate Approximation (PAA) window size of 16 and an alphabet size of 8.
- **Bag-of-SFA-Symbols (BOSS)** (Schäfer, 2015): A time series classification algorithm that uses Symbolic Fourier Approximation (SFA) to generate symbolic words (tokens). This was used as a baseline in the quantitative finance domain, applied to the mid-price series.
- **QualTok (Ablation Baseline)**: As described in Section 5, QualTok serves as an ablation baseline for QA-Token. It employs a simplified quality-aware merge score, $w_{ab} \propto \frac{f(a,b)}{f(a)f(b)+\epsilon_f} \cdot \left(\frac{q_a+q_b}{2} + \epsilon_Q\right)^{\alpha}$, but critically omits the reinforcement learning policy optimization for merge sequences and the full adaptive learning loop for complex θ_{adapt} parameters beyond tuning α. Merge operations are typically performed greedily based on this score.

For all baseline methods, we select essential hyperparameters, such as the target vocabulary size (which typically corresponds to a predefined number of merge operations, e.g., 16,000 or 32,000, as specified per domain in Section 5), based on common practices in the literature (Sennrich et al., 2016; Kudo & Richardson, 2018; Wu et al., 2016; Devlin et al., 2019; Brown et al., 2020; Ji et al., 2021), specific recommendations from the original implementations of these methods, or by identifying the best-performing configuration on a held-out validation set from a systematic sweep of reasonable values to ensure robust comparisons.

I.5. Evaluation Metrics

The performance of QA-Token and baseline methods was assessed using the following domain-specific metrics, corresponding to the results presented in Section 5.

- **Genomics:**
  - **Variant Calling:** Performance was measured by F1-score, precision, and recall against the GIAB truth sets. These metrics were computed using the 'hap.py' tool (version 0.3.14), available at `https://github.com/Illumina/hap.py`.
  - **Taxonomic Classification (Metagenomics):** For the CAMI II benchmark, performance was primarily assessed using classification accuracy (specifically, the F1-score for overall classification performance, as reported in Table 1).
  - **Sequence Reconstruction Loss:** The quality of token representations was also evaluated by training Transformer-based autoencoder models and measuring the reconstruction loss (e.g., cross-entropy for discrete tokens) on a held-out test set.

  Variant Calling Model Architecture: The variant calling evaluation uses a Transformer encoder that takes token embeddings as input features. The model outputs per-position variant probabilities (SNV, insertion, deletion, reference). Training uses cross-entropy loss against GIAB HG002 labels. This approach evaluates how well tokenization preserves variant-informative sequence features in the learned representations, with evaluation performed using the `hap.py` benchmarking tool (v0.3.14).

- **Quantitative Finance:**
  - **Return Prediction Accuracy:** The percentage of correctly predicted signs for future (e.g., 5-minute ahead) mid-price returns.
  - **Volatility Forecasting RMSE:** The Root Mean Squared Error between the predicted 5-minute volatility and the realized volatility (computed from higher-frequency data).
  - **Market Regime Identification Accuracy:** The accuracy achieved in classifying time periods into discrete market states (e.g., two states identified by a GARCH-HMM).
  - **Trading Performance:** The primary metric was the annualized Sharpe Ratio (Sharpe, 1994) achieved by a PPO-based trading agent operating on the tokenized data. A transaction cost of 5 basis points per trade was incorporated. Additional performance metrics, such as Maximum Drawdown (MDD) and Calmar Ratio, were also monitored (see Appendix H.4 for further details).

- **Social Media Text:**
  - Performance on the seven TweetEval benchmark tasks was measured using the official evaluation metric specified by the benchmark organizers for each respective task (Barbieri et al., 2020). These metrics are:
    - Emoji Prediction: Accuracy (Acc)
    - Emotion Recognition: Macro F1-score (F1 M)
    - Hate Speech Detection: Macro F1-score (F1 M)
    - Irony Detection: Accuracy (Acc)
    - Offensive Language Identification: Macro F1-score (F1 M)
    - Sentiment Analysis: Macro Recall (Rec M)
    - Stance Detection: Average F1-score across topics (F1 Avg)

All reported experimental results in Section 5 represent the mean and 95% confidence interval over $n = 10$ independent runs to ensure robustness and allow for assessment of variability.

I.6. Code Availability

We will release the QA-Token framework on GitHub under an MIT license. The repository includes source code, configuration files, pre-trained models, and reproducibility scripts for all experiments.

I.7. Approximating QA-Token: Towards Computationally Efficient Quality-Awareness

The learning framework of QA-Token has high computational costs due to both RL and adaptive learning stages. Future work will explore computationally lighter approximations. A starting point is our ablation baseline, QualTok, which uses a greedy merge strategy based on the quality-aware score w_{ab} (Equation 5) without explicit RL policy optimization, bypassing the costs of Stage 1 RL.

Further cost reduction can be achieved by:

1. **Streamlined Adaptive Parameter Learning for Greedy Merges:** Instead of full RL, we can focus on adaptively learning a refined set of parameters θ^*_{adapt} (e.g., α, quality weights w_j, simplified reward weights λ_j) that directly optimize the greedy w_{ab}-guided tokenization for downstream tasks. This retains the core quality-aware adaptability while significantly reducing complexity compared to learning an RL policy. The Gumbel-Softmax based learning (Stage 2) would optimize θ_{adapt} for these greedy merges, possibly using simplified composite logits.

2. **Policy Distillation:** If the RL policy $\pi^*_{\theta_\pi}$ captures complex merge dependencies, the computational overhead at deployment can be mitigated. A compact "student" model (e.g., a smaller neural network or decision tree) can be trained via policy distillation (Hinton et al., 2015; Rusu et al., 2016) to mimic the decisions of a larger, pre-trained "teacher" RL agent, offering faster vocabulary construction.

3. **Surrogate-Assisted Adaptive Learning:** The optimization of θ_{adapt} (Stage 2) can be accelerated by using cheaper-to-evaluate surrogate models (Jones et al., 1998) to approximate the downstream task loss L_{task}, reducing the need for frequent, costly end-to-end evaluations with the full downstream model.

4. **Transfer and Meta-Learning for θ_{adapt}:** Leveraging learned θ_{adapt} parameters from one task or dataset as initializations for others (as in Algorithm 6) can substantially reduce the training burden for new applications.

I.8. Extended TweetEval Benchmarking Methodology

This section describes the comprehensive TweetEval benchmarking methodology. Results are reported in Table 21.

Datasets and Evaluation Framework: TweetEval (Barbieri et al., 2020) provides a unified framework for evaluating models on seven heterogeneous tweet classification tasks, each with fixed training, validation, and test splits. This allows for standardized comparison across different approaches. The seven tasks are: Emotion Recognition (Mohammad et al., 2018) (4 labels: anger, joy, sadness, optimism), Emoji Prediction (Barbieri et al., 2018) (20 emoji labels), Irony Detection (Van Hee et al., 2018) (2 labels: irony, not irony), Hate Speech Detection (Basile et al., 2019) (2 labels: hateful, not hateful), Offensive Language Identification (Zampieri et al., 2019) (2 labels: offensive, not offensive), Sentiment Analysis (Rosenthal et al., 2017) (3 labels: positive, neutral, negative), and Stance Detection (Mohammad et al., 2016) (3 labels: favour, neutral, against, across five topics). For each task, we report performance using the unified evaluation metrics specified by the TweetEval benchmark. Table 21 provides the baseline comparison framework. The official metric for each task as defined by TweetEval (also see *https://github.com/cardiffnlp/tweeteval* for details) is reported.

Table 21. TweetEval Baseline Comparison Framework.

Model	Emoji	Emotion	Hate	Irony	Offensive	Sentiment	Stance	ALL(TE)
BERTweet	33.4	79.3	**56.4**	**82.1**	79.5	73.4	71.2	**67.9**
TimeLMs-2021	**34.0**	**80.2**	55.1	64.5	**82.2**	**73.7**	**72.9**	66.2
RoBERTa-Retrained	31.4	78.5	52.3	61.7	80.5	72.8	69.3	65.2
RoBERTa-Base	30.9	76.1	46.6	59.7	79.5	71.3	68.0	61.3
RoBERTa-Twitter	29.3	72.0	49.9	65.4	77.1	69.1	66.7	61.4
FastText	25.8	65.2	50.6	63.1	73.4	62.9	65.4	58.1
LSTM	24.7	66.0	52.6	62.8	71.7	58.3	59.4	56.5
SVM	29.3	64.7	36.7	61.7	52.3	62.9	67.3	53.5
QA-BPE-nlp + BERTweet	**34.2**	**81.5**	**58.8**	**82.9**	**83.0**	**75.1**	**73.5**	**70.0**