**RoastFormer: Transformer-Based Time-Series Generation for Coffee Roast Profiles**

**1. Project Title and Overview**

**RoastFormer** explores decoder-only transformer architectures for continuous time-series generation using coffee roast profiles as a case study.

Domain Context: Coffee roasters currently develop roast profiles through iterative trial-and-error combined with sensory cupping—tasting coffees to evaluate how time-temperature curves affect flavor outcomes. While modern roasters use data logging software (like Artisan or Cropster) to visualize and replicate roast curves as temperature-over-time graphs, there is no predictive starting point: each new coffee requires building a profile from scratch through experimentation (10-20 experimental roasts - each ~15 minutes plus cupping time). Roasters manipulate heat application based on sight (bean color), smell (aromatic development), and sound ("first crack"), then taste the result through standardized cupping protocols. Once a successful profile is discovered, the logged curve becomes a repeatable recipe, but the initial profile creation remains an expertise-driven, time-intensive process requiring dozens of experimental roasts and extensive cupping experience to understand the flavor space.

Project Approach: **RoastFormer** investigates whether transformers designed for discrete text can generate physically plausible continuous time-series profiles conditioned on bean characteristics (origin, density, processing method) and desired outcomes (light/medium/dark roast levels). I treat roasting as an autoregressive sequence modeling problem where temperature readings are generated token-by-token, analogous **to GPT's text generation u**sing **physics-based synthetic data validated against real Artisan CSV exports**,

**2. Connection to Course Material**

Direct implementation of Phuong & Hutter's formal algorithms: (1) **Tokenizing continuous data**—discretizing temperatures into vocabulary tokens, adapting NLP tokenization to physical measurements; (2) **Positional encoding for temporal sequences**—comparing learned vs. sinusoidal vs. RoPE (my paper presentation), where temporal relationships are critical; (3) **Causal attention masking**—preventing future information leakage essential for real-time prediction; (4) **Architecture ablations**—empirically testing which components transfer from discrete language to continuous time-series. Goes beyond "using AI" by investigating why architectural choices succeed or fail when adapting NLP transformers to physics-constrained temporal data.

**3. Problem Statement and Goals**

**Problem**: Can transformers, designed for discrete language tokens, generate physically plausible starting-point profiles for continuous temporal sequences with physical constraints? Which transformer architectural components (positional encodings, attention heads, layer depth) are critical for time-series generation versus language modeling? Does the model learn interpretable roasting concepts (heat-up rate, development time ratio) or merely fit curves?

**Goals**:

(1) Adapt decoder-only transformer for continuous temperature data; (2) Compare positional encoding methods on generation quality; (3) Analyze whether attention learns roasting phases without supervision; (4) Validate synthetic profiles against real Artisan exports; (5) Identify which components matter through systematic ablations.

**Success**: Generate profiles matching real Artisan characteristics (3-second intervals, 10-15 minute duration, 70-450°F range, smooth phase transitions) with <5°F average error—demonstrating good data science practice through proper evaluation over raw performance. I have 2 real Artisan CSV outputs at this time.

**4. Timeline**

| **Week** | **Milestone** | **Deliverable** |
| --- | --- | --- |
| Oct 20-25 | Project definition | Proposal submitted, discussed with TA, proposal submitted |
| Oct 27-Nov 1 | Data generation | Generate 10K synthetic profiles validated against real Artisan exports, establish baselines |
| Nov 3-8 | Base implementation | Implement DTransformer working with 3 positional encoding variants |
| Nov 10-15 | Experiments & analysis | Run ablations, analyze attention patterns, create visualizations |
| Nov 17-20 | Final prep | Presentation materials, Model Card, complete pseudocode |

**5. Success Metrics**

**Quantitative**: (1) Temperature prediction MAE <5°F (practical roasting tolerance); (2) Physical plausibility: 100% monotonic increase, >95% bounded heating rates, >90% reach target temps; (3) **Extrapolation test** (key differentiator): Generate 20-min profiles after training on 15-min—which positional encoding maintains plausibility?

**Qualitative**: (1) Attention pattern analysis—do heads specialize on roast phases?; (2) Conditioning fidelity—do dense beans generate 20-25% slower heating as physics predicts?; (3) Ablation rankings showing component importance.

**Baselines**: Linear interpolation, LSTM (same parameters), physics-only simulation. Success requires understanding *why* components matter, not just higher performance.

**6. Value Proposition**

**For roasters**: Demonstrates proof-of-concept for data-driven starting profiles that could reduce experimental iterations and coffee waste, though this theoretical work with synthetic data would require real-data extension and should augment rather than replace roaster expertise.

**For practitioners**: Complete pseudocode, synthetic data generation framework, and documented design decisions offer reusable blueprint for transformer-based time-series generation in any domain (industrial control, climate modeling, financial forecasting).

**Note**: This is a more so theoretical/pseudocode-focused project as compute and especially data is limited (highly protected by individual roasters. The emphasis is on understanding transformer architectures through rigorous adaptation to a new domain, with potential future extension to real roasting data through partnerships with roasteries using modern logging software.

formal alg, this, create repo and read me, ideal for the use by claude code

Use CLAUDE CODE – do an anaylsis of all this, and look at repo

Take repo, come up with a plan, do not start work

Best practice for Git up flow – treat claude code like a team member

Issues to repo, branch off, test it, then go back inot main

What measure for accuracy? Add new timeline, reduce to one page

Less waste, good for

**RoastFormer: Transformer-Based Time-Series Generation for Coffee Roast Profiles**

**1. Project Title and Overview**

Coffee roasters develop profiles through trial-and-error: each new coffee requires 10-20 experimental roasts (~15 minutes each plus cupping time) to build a temperature-over-time curve from scratch. Modern logging software (Artisan, Cropster) captures and replicates successful profiles, but no predictive starting point exists—roasters must experiment from zero for every coffee, manipulating heat based on sight, smell, and sound, then evaluating through cupping protocols. **RoastFormer** investigates whether transformers designed for discrete text can generate physically plausible continuous time-series profiles conditioned on bean characteristics (density, origin, processing) and desired outcomes (light/medium/dark). Using physics-based synthetic data validated against real Artisan CSV exports, I implement Phuong & Hutter's DTransformer, treating roasting as autoregressive sequence modeling where temperature readings are generated token-by-token analogous to GPT's text generation.

**2. Connection to Course Material**

Direct implementation of Phuong & Hutter's formal algorithms: (1) tokenizing continuous temperatures into vocabulary tokens, adapting NLP tokenization to physical measurements; (2) comparing positional encoding methods (learned, sinusoidal, RoPE from my paper presentation) where temporal relationships are critical; (3) causal attention masking preventing future information leakage; (4) architecture ablations empirically testing which components transfer from discrete language to continuous time-series. Goes beyond "using AI" by investigating *why* architectural choices succeed or fail when adapting NLP transformers to physics-constrained temporal data.

**3. Problem Statement and Goals**

**Problem**: Can transformers designed for discrete tokens generate physically plausible continuous temporal sequences? Which architectural components (positional encodings, attention, depth) are critical for time-series versus language? Does the model learn interpretable roasting concepts or merely fit curves? **Goals**: (1) Adapt decoder-only transformer for continuous temperature data; (2) compare positional encoding methods; (3) analyze if attention learns roasting phases without supervision; (4) validate synthetic profiles against real Artisan exports; (5) identify component importance through ablations. **Success**: Generate profiles matching real Artisan characteristics (3-second intervals, 10-15 minutes, 70-450°F, smooth transitions) with <5°F average error, demonstrating good data science practice over raw performance.

**4. Timeline**

**Oct 20-25**: Project definition, proposal submission. **Oct 27-Nov 1**: Generate 10K synthetic profiles validated against 2 real Artisan exports, establish baselines. **Nov 3-8**: Implement DTransformer with 3 positional encoding variants. **Nov 10-15**: Run ablations, analyze attention, create visualizations. **Nov 17-20**: Finalize presentation (real vs. generated comparison), Model Card, pseudocode.

**5. Success Metrics**

**Quantitative**: (1) Mean Absolute Error <5°F across profiles (practical roasting tolerance); (2) Structural match to Artisan data—3-second sampling, realistic heating rates (20-100°F/min); (3) Physical plausibility—100% monotonic increase, >95% bounded rates, >90% reach targets. **Qualitative**: (1) Do attention heads specialize on roast phases without explicit training?; (2) Do dense beans generate 20-25% slower heating as physics predicts?; (3) Do generated curves visually resemble real Artisan profiles? **Baselines**: Linear interpolation, LSTM. Success means understanding *which* components enable time-series generation, not just scores.

**6. Value Proposition**

**For roasters**: Proof-of-concept for data-driven starting profiles reducing experimental iterations, validated against real Artisan exports though requiring real-data extension for deployment. **For practitioners**: Reusable pseudocode and synthetic data framework validated against real-world data, demonstrating transformer adaptation to continuous time-series generation across domains (industrial control, climate, finance).