E Ethical Optimization and Monitoring Framework

Implementation Blueprint based on the
E Mathematics & Structure Model

October 2025

1 Core Mathematical Definitions

Define the ethical delta as:

$$\Delta E(t) = 1 - \frac{V_{CoT}(t) \cdot V_{Ideal}}{\|V_{CoT}(t)\| \|V_{Ideal}\|}$$

The unified loss couples task performance and ethics:

$$L_{total} = L_{task} + \lambda \, \Delta E(t)$$

Alternatively, the ratio-based moral scaling form:

$$S(t) = \frac{L_{task}(t)}{(\Delta E(t) + \varepsilon)^n}$$

Temporal integral for sustained ethical optimization:

$$J = \int_{t_0}^{t_T} \frac{L_{task}(t)}{(\Delta E(t))^n} \, dt$$

Optional stability penalty to promote smooth ethical trajectories:

$$L_{stab} = \beta \, \|\nabla_t \Delta E(t)\|^2$$

2 Pipeline Overview

1. Generate latent embeddings:
   - $V_{CoT}(t)$: reasoning embedding.
   - V_{Ideal}: ethical anchor vector.
2. Compute Semantic Delta:

$$\Delta_s = \|E_t - E_0\|_2$$

3. Use an Ethics SVM for classification of ethical alignment.
4. Compute $\Delta E(t)$ from cosine dissimilarity.
5. Combine metrics and apply as loss or monitoring term.
6. Log metrics $\{L_{task}, \Delta E, \Delta_s, D_e\}$ for visualization and review.

3 Python-Style Pseudocode

Listing 1: Core Implementation Outline

```
%%
%% This is file `rename-to-empty-base.tex',
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% fileerr.dtx  (with options: `return')
%%
%% This is a generated file.
%%
%% The source is maintained by the LaTeX Project team and bug
%% reports for it can be opened at https://latex-project.org/bugs/
%% (but please observe conditions on bug reports sent to that address!)
%%
%%
%% Copyright (C) 1993-2025
%% The LaTeX Project and any individual authors listed elsewhere
%% in this file.
%%
%% This file was generated from file(s) of the Standard LaTeX `Tools
   Bundle'.
%%
   ------------------------------------------------------------------------

%%
%% It may be distributed and/or modified under the
%% conditions of the LaTeX Project Public License, either version 1.3c
%% of this license or (at your option) any later version.
%% The latest version of this license is in
%%    https://www.latex-project.org/lppl.txt
%% and version 1.3c or later is part of all distributions of LaTeX
%% version 2005/12/01 or later.
%%
%% This file may only be distributed together with a copy of the LaTeX
%% `Tools Bundle'. You may however distribute the LaTeX `Tools Bundle'
%% without such generated files.
%%
%% The list of all files belonging to the LaTeX `Tools Bundle' is
%% given in the file `manifest.txt'.
%%
```

```
 \message{File ignored}
\endinput
%%
%% End of file `rename-to-empty-base.tex'.
```

```
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

EPS = 1e-8

def cosine_dissimilarity(a, b):
    na = np.linalg.norm(a) + EPS
    nb = np.linalg.norm(b) + EPS
    return 1.0 - float(np.dot(a, b) / (na * nb))

def l2(a, b):
    return float(np.linalg.norm(a - b))

class EthicsSVM:
    def __init__(self):
        self.scaler = StandardScaler()
        self.svm = SVC(kernel='rbf', probability=True)
    def fit(self, X, y):
        self.svm.fit(self.scaler.fit_transform(X), y)
    def score_and_distance(self, x):
        xs = self.scaler.transform([x])
        dist = self.svm.decision_function(xs)[0]
        div = 1.0 - (1/(1+np.exp(-dist)))
        return np.clip(div, 0.0, 1.0)

class EthicalMonitor:
    def __init__(self, V_ideal, E0, svm,
                 sem_delta_thresh=0.18,
                 eth_div_thresh=0.22,
                 consecutive_trigger=3,
                 ema_alpha=0.2):
        self.V_ideal = V_ideal
        self.E0 = E0
        self.svm = svm
        self.sem_delta_thresh = sem_delta_thresh
        self.eth_div_thresh = eth_div_thresh
        self.consec = 0
        self.ema_alpha = ema_alpha
        self.ema_DeltaE = None
        self.log = []

    def process_output(self, E_t, V_CoT, L_task):
        delta_s = l2(E_t, self.E0)
        deltaE = cosine_dissimilarity(V_CoT, self.V_ideal)
        svm_div = self.svm.score_and_distance(np.concatenate([E_t, V_CoT])
           )
        combined_div = 0.6*deltaE + 0.3*svm_div + 0.1*delta_s
```

```
        if self.ema_DeltaE is None:
            self.ema_DeltaE = combined_div
        else:
            self.ema_DeltaE = self.ema_alpha*combined_div + \
                              (1-self.ema_alpha)*self.ema_DeltaE

        trigger = (combined_div > self.eth_div_thresh) \
                  or (delta_s > self.sem_delta_thresh)
        if trigger:
            self.consec += 1
        else:
            self.consec = 0

        action = 'ok'
        if self.consec >= 3:
            action = 'flag_for_review'

        self.log.append(dict(
            L_task=L_task, deltaE=deltaE,
            delta_s=delta_s, svm_div=svm_div,
            combined_div=combined_div, ema=self.ema_DeltaE,
            action=action))
        return action
```

4 Thresholds and Parameters

- Semantic delta threshold: $\Delta_s \in [0.15, 0.25]$
- Ethical divergence threshold: $\Delta E \in [0.2, 0.3]$
- Consecutive trigger count: 3
- EMA smoothing parameter: $\alpha = 0.2$

5 Training Procedure

1. Build V_{Ideal} by averaging embeddings of curated aligned corpora.
2. Train Ethics SVM on labeled examples (aligned / misaligned).
3. Integrate $\lambda \Delta E$ into loss during fine-tuning.
4. Deploy inference-time monitor to compute ΔE, SVM divergence, and Δ_s per output.
5. Log all data for analysis and audit.

6 Experiment Design

- Perform grid search on λ, n, and β.
- Measure task performance, ethical stability, and long-term drift.
- Evaluate using:

$$S_{eval} = \frac{\text{task-score}}{(\overline{\Delta E} + \varepsilon)^n}$$

7 Explainability and Human Oversight

When flagged:

- Report cause of flag (which signal triggered).
- Provide a human-readable explanation such as:

  "Output diverged by 0.34 ΔE from ethical anchor; recommend review."

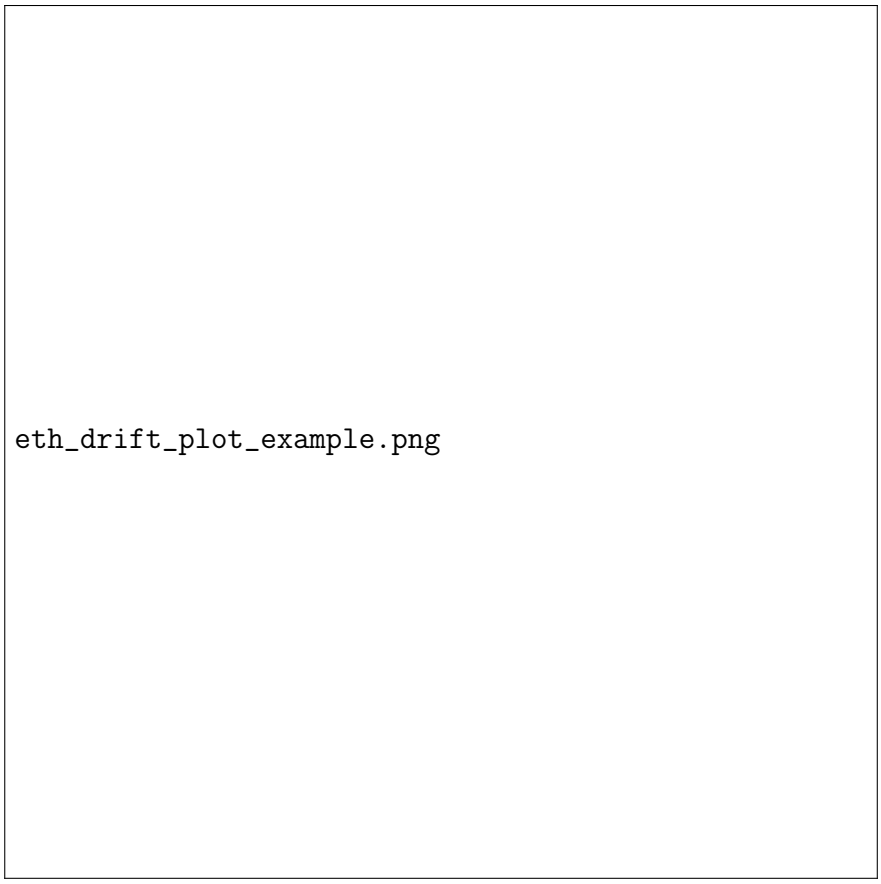
- Include corrective suggestions for model or operator.

8 Safety and Governance

- Protect embedding privacy (store hashed or aggregated forms).
- Maintain versioned ethical anchors and document curatorial sources.
- Require human review for critical outputs.
- Monitor for metric gaming or surface-level compliance.

9 Visualization and Metrics

- Plot time series of ΔE, Δ_s, and combined divergence.
- Visualize ethical trajectory using:



- Use 2D scatter (Semantic Δ vs. ΔE) to observe drift clusters.

10 Roadmap

1. Construct and validate ethical anchor V_{Ideal}.
2. Collect labeled ethics data and train SVM.
3. Deploy `EthicalMonitor` in shadow mode for observation.
4. Tune thresholds and coefficients.
5. Enable staged interventions with human-in-loop review.

Implementation blueprint derived from the E Ethics Optimization model (Beckett, 2025).