PAEC kNN-MT: Stabilizing Retrieval-Augmented Translation via Lyapunov-Guided Control

Anonymous ACL submission

Abstract

k-Nearest Neighbor Machine Translation (kNN-MT) enables effective domain adaptation but suffers from greedy per-step decisions that ignore long-term consequences, amplifying *exposure bias* as early errors compound throughout translation. We propose **PAEC** (**P**roduction-**A**ware **E**xposure **C**ontrol), which reframes kNN-MT retrieval control as a dynamical system stabilization problem. PAEC models decoding as an 11-dimensional state evolution and learns a Transformer-based dynamics model $\mathcal{T}_\theta$ with provable *Lipschitz continuity*. By enforcing *Control Lyapunov Function* conditions, we derive policies guaranteeing *Ultimate Uniform Boundedness*: errors converge to a bounded neighborhood with high probability. On De-En translation under constrained settings (enforced retrieval, limited datastore), PAEC reduces the failure rate from 34.2% to 29.3% ($p < 10^{-12}$). While Adaptive kNN-MT degenerates to Pure NMT ($\lambda \to 0$), PAEC stabilizes the retrieval pathway, demonstrating tail-risk avoidance. Finally, we distill the online controller into an offline policy, achieving 93% latency reduction while preserving comparable stability guarantees. As a control layer orthogonal to retrieval mechanisms, PAEC enables integration with other kNN-MT variants.

1 Introduction

k-Nearest Neighbor Machine Translation (kNN-MT; Khandelwal et al. 2021) enables non-parametric domain adaptation by interpolating NMT outputs with distributions retrieved from an external datastore. However, per-step retrieval from large-scale datastores incurs prohibitive latency, motivating research into adaptive variants.

Adaptive methods (Zheng et al., 2021) learn when to retrieve based on model confidence. While effective at reducing latency, they employ *greedy single-step optimization*—maximizing immediate gain without modeling long-term outcomes. In autoregressive generation, suboptimal decisions can amplify *exposure bias* (Ranzato et al., 2016; Wang and Sennrich, 2020), where early errors compound throughout translation. Existing methods lack formal guarantees against such cascading risks.

Furthermore, a critical observation emerges: a suboptimal datastore can cause adaptive policies to degenerate ($\lambda \to 0$), forfeiting non-parametric adaptation, which leads to our new question: **can retrieval be stabilized rather than bypassed?** To validate this statement, our research conducts a **forced retrieval regime**: retrieval remains mandatory despite datastore imperfections, shifting the objective from circumventing unreliable retrieval to ensuring its stability via formal control.

We propose **PAEC** (**P**roduction-**A**ware **E**xposure **C**ontrol), which reframes kNN-MT retrieval control as a *dynamical system stabilization problem*. The decoding process is modeled as a state evolution where, in each step, the retrieval setting is a control input for state transition. By applying Lyapunov stability theory (Lyapunov, 1992; Khalil, 2002), we derive policies with provable error bounds and controllable kNN retrievals.

Our contributions:

- We formalize kNN-MT decoding as an 11D-state dynamical system and prove our dynamics model $\mathcal{T}$ is *Lipschitz continuous* (Theorem C.2), ensuring bounded sensitivity to perturbations.
- We design a quadratic Lyapunov function $V(\mathcal{E}) = \mathcal{E}^\top \mathbf{P} \mathcal{E}$ (Definition D.4) and enforce *Control Lyapunov Function* conditions during training (Definition D.3; Artstein 1983; Sontag 1989), achieving *Ultimate Uniform Boundedness* (Theorem 3.2): errors converge to a bounded neighborhood with high probability.
- PAEC operates as a *control layer* orthogonal to

the retrieval mechanism, governing *when* and *how* to retrieve, thereby enabling integrations with other kNN-MT variants.

- We distill the online controller into an offline policy, achieving 93% latency reduction (27.7s $\rightarrow$ 1.9s per sentence). On OPUS-100/Europarl De-En under forced retrieval (constrained 50K-entry datastore), PAEC reduces failure rate from 34.2% to 29.3% vs. Vanilla kNN-MT $\left(p < 10^{-12}\right)$. Under these conditions, Adaptive kNN-MT degenerates to Pure NMT (BLEU: $36.68 \approx 36.60$) by abandoning retrieval, while PAEC maintains active retrieval with bounded error accumulation.

2 Background

2.1 kNN-MT Preliminaries

kNN-MT (Khandelwal et al., 2021) augments the NMT decoder by retrieving from a datastore $\mathcal{D} = \{(\mathbf{k}_i, v_i)\}$, where $\mathbf{k}_i$ is a context representation and v_i is the corresponding target token. At step t, the decoder hidden state serves as a query $\mathbf{q}_t$ to retrieve k nearest neighbors $\mathcal{N}_k$:

$$p_{k\text{NN}}\left(y_t\right) \propto \sum_{(\mathbf{k}_i, v_i) \in \mathcal{N}_k} \mathbb{1}\left[v_i = y_t\right] \cdot \exp\left[-\frac{d\left(\mathbf{q}_t, \mathbf{k}_i\right)}{T}\right]$$

The final prediction interpolates NMT and kNN distributions:

$$p\left(y_t\right) = \lambda \cdot p_{k\text{NN}}\left(y_t\right) + (1 - \lambda) \cdot p_{\text{NMT}}\left(y_t\right)$$

2.2 Limitations of Adaptive Retrieval

Adaptive kNN-MT (Zheng et al., 2021) trains a meta-network to dynamically select k for noise filtering, but suffers from two key limitations:

- **Myopic Optimization:** The meta-network is optimized for immediate probability gain, ignoring the long-term impact of current retrieval actions. This greediness can compound semantic drift autoregressively.
- **Unbounded Error Propagation:** The framework lacks formal stability guarantees. In high-risk scenarios, the heuristic may skip retrieval based on transient confidence, allowing errors to cascade unchecked.

2.3 Control-Theoretic Formulation

We formulate kNN-MT decoding as a discrete-time dynamical system where the state $\mathbf{S}_t \in \mathbb{R}^{11}$ evolves under retrieval actions $\mathbf{A}_t$. The control objective is to minimize cumulative error energy under resource constraints, transforming retrieval scheduling from greedy heuristics to formally stable optimal control (see Section 3).

Connection to Exposure Bias Exposure bias compounds errors in autoregressive generation. PAEC does not eliminate the bias but bounds its *consequences*: the Lyapunov constraint on the error state $\mathcal{E}_t$ (Section 3.1) provides formal guarantees for boundedness of these compounding errors irrespective of trajectory length.

3 Method: The PAEC Framework

We present **PAEC** (**P**roduction-**A**ware **E**xposure **C**ontrol), a control-theoretic framework for stabilizing kNN-MT systems, where the "production-aware" aspect incorporates resource constraints (latency, memory, throughput) alongside translation quality. Our approach reformulates decoding as a discrete-time dynamical system and employs Lyapunov stability theory to learn policies that provably bound translation errors.

3.1 State Space Design

The system state at step t is an 11-dimensional vector capturing the decoding status:

Definition 3.1 (Total System State)**.** *The system state $\mathbf{S}_t \in \mathcal{S}$ is defined as:*

$$\mathbf{S}_t = \left[\mathcal{E}_t, \Phi_t, \mathbf{H}_t\right]^\top \in \mathbb{R}^{11}, \tag{3.1}$$

where:

- $\mathcal{E}_t \in \mathbb{R}^4$*:* ***Error state*** *quantifying translation quality deviation in semantic drift, entity coverage, local fluency (surprisal), and repetition;*
- $\Phi_t \in \mathbb{R}^3$*:* ***Resource pressure*** *capturing normalized stress on latency, memory, and throughput relative to SLA constraints;*
- $\mathbf{H}_t \in \mathbb{R}^4$*:* ***Generative context*** *encoding the decoder's internal state: attention focus, alignment consistency, representation stability, and confidence volatility.*

Remark 3.1 (Properties of System State)**.** *The state space is designed to satisfy:*

(1). Approximate Markovian sufficiency—$\mathbf{S}_t$ *captures decision-critical information;* $\mathcal{T}_\theta$ *conditions on a* w*-step history window for accuracy (Definition B.2).*
(2). Observability—*all components are computable at runtime.*
(3). Compactness—*bounded values ensure numerical stability.*

Formal definitions are in Appendix A.

Error State $\mathcal{E}_t$. Each component measures a distinct quality aspect:

- Semantic drift $\epsilon_t^{(\text{sem})} \in [0, 2]$ via embedding cosine distance (Definition A.2);
- Entity coverage $\epsilon_t^{(\text{cov})} \in [0, 1]$ tracking untranslated named entities (Definition A.3);
- Surprisal $\epsilon_t^{(\text{surp})} \in [0, 1]$ from language model perplexity (Definition A.4);
- Repetition $\epsilon_t^{(\text{rep})} \in [0, 1]$ detecting n-gram loops (Definition A.5).

Pressure State $\mathbf{\Phi}_t$. Sigmoid normalization maps physical measurements—latency L_t, memory M_t, and throughput R_t—to the bounded interval $(\epsilon, 1-\epsilon)$ (Definition A.6; Proposition A.3). Throughput pressure employs an asymmetric design, penalizing under-capacity more severely than over-capacity.

Context State $\mathbf{H}_t$. This vector captures the decoder's "mental state" (Definition A.7; Proposition A.4): attention focus on uncovered entities, query-context alignment, cross-step trajectory stability, and confidence volatility.

Hybrid Normalization $\mathbf{\Psi}$. A bijective hybrid transformation $\Psi : \mathbb{R}^{11} \to \mathbb{R}^{11}$ (Appendix A) ensures numerical stability for the heterogeneously distributed raw state components. It applies `StandardScaler` to normally distributed components and `QuantileTransformer` to skewed ones, mapping all inputs to an approximate standard normal distribution while preserving Lipschitz continuity.

3.2 Dynamics Model $\mathcal{T}_\theta$

We model the kNN-MT system as a discrete-time dynamical system:

$$\mathbf{S}_{t+1} = \mathcal{F}\left(\mathbf{S}_t, \mathbf{A}_t\right) + \boldsymbol{\xi}_t \tag{3.2}$$

where $\mathbf{A}_t \in \mathbb{R}^6$ is the action vector (one-hot index type, continuous k and λ), and $\boldsymbol{\xi}_t$ represents system noise.

Conditional Modeling & Delta Prediction. Our dynamics model $\mathcal{T}_\theta$ is designed with two key insights:

- The pressure state Φ_t is *exogenous*, reflecting external system factors beyond the control of retrieval decisions. The model conditions on observed Φ_t but predicts only internal states $(\mathcal{E}, \mathbf{H})$, avoiding data leakage from unpredictable environmental dynamics.
- To capture local state evolution and reduce learning difficulty, $\mathcal{T}_\theta$ predicts the *state increment* rather than absolute values (Definition B.10):

$$\left[\Delta\hat{\mathcal{E}}_t, \Delta\hat{\mathbf{H}}_t\right] = \mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}_t; \mathbf{S}_{\text{hist}}, \mathbf{A}_{\text{hist}}\right) \tag{3.3}$$

  where $(\mathbf{S}_{\text{hist}}, \mathbf{A}_{\text{hist}})$ denotes the historical window of w steps (Definition B.2). This residual formulation $\left(\text{e.g., } \mathcal{E}_{t+1} = \mathcal{E}_t + \Delta\hat{\mathcal{E}}_t\right)$ naturally enforces trajectory continuity.

Architecture. Our dynamics model $\mathcal{T}_\theta$ is a Transformer encoder that processes a $(w+1)$-step state-action sequence (w historical steps plus the current step):

1. **Sequence embedding:** Historical state-action pairs $\{(\mathbf{S}_{t-w+1}, \mathbf{A}_{t-w+1}), \ldots, (\mathbf{S}_t, \mathbf{A}_t)\}$ are projected via MLPs and augmented with sinusoidal positional encodings.
2. **Transformer encoding:** The Transformer layer employs multi-head self-attention to capture temporal dependencies, followed by feedforward networks with SiLU activation.
3. **Action fusion:** The current action embedding is fused with the encoded state representation via cross-attention.
4. **Prediction heads:** Separate MLP heads predict the next error state $\hat{\mathcal{E}}_{t+1}$ and auxiliary term $\hat{\mathbf{H}}_{t+1}$; we primarily use delta prediction to better model local changes.

Detailed specifications are in Appendix B.

3.3 Theoretical Guarantees

PAEC provides formal stability guarantees through two complementary analyses, with Lipschitz continuity and Lyapunov stability to ensure robust control.

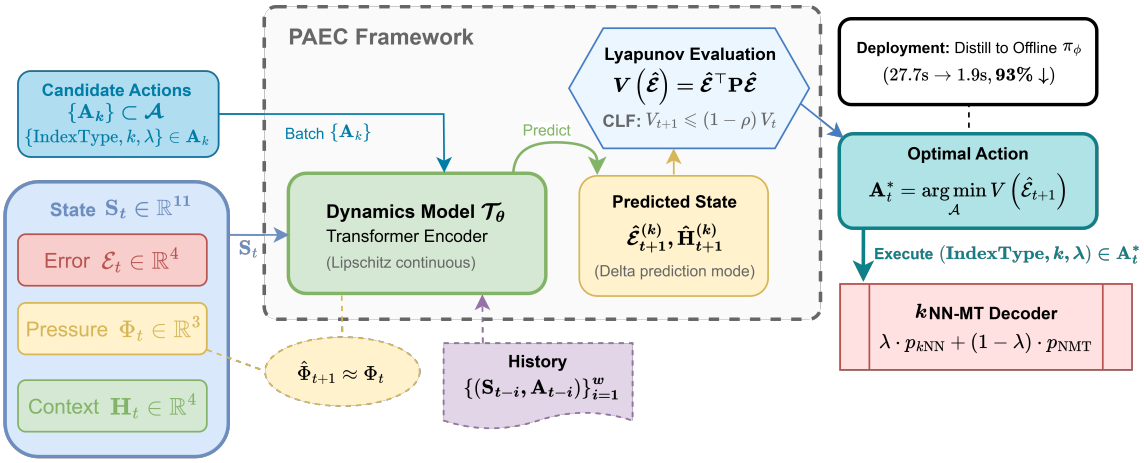

Figure 1: Overview of the PAEC framework. The Transformer-based dynamics model $\mathcal{T}_\theta$ predicts next-step internal states $(\mathcal{E}_{t+1}, \mathbf{H}_{t+1})$ conditioned on current state $\mathbf{S}_t$ and action $\mathbf{A}_t$. The implicit policy selects actions minimizing the Lyapunov function $V(\mathcal{E})$.

Lipschitz Continuity. We require that small input perturbations yield bounded output changes. Under spectral normalization, the dynamics model $\mathcal{T}_\theta$ satisfies the following (proof in Appendix C):

Theorem 3.1 (Local Lipschitz Property of $\mathcal{T}_\theta$)**.** *Under spectral normalization of all linear layers, the dynamics model $\mathcal{T}_\theta$ satisfies:*

(i) a theoretical Lipschitz upper bound $L_{theory} \approx 126$ on the full input space;
(ii) an empirical local Lipschitz constant $L_{emp} \approx 1.02$ on the training distribution, with $P_{99} \approx 1.09$.

The gap between L_{theory} and L_{emp} arises as the preprocessing transformation Ψ exhibits high curvature primarily at the distribution tails (Remark C.3). For stability analysis, we use L_{emp} to govern typical behavior, while L_{theory} conservatively bounds out-of-distribution inputs.

This near-unity constant of L_{emp} ensures $\mathcal{T}_\theta$ is *almost non-expansive*, preventing error amplification during iterative prediction. We achieve this mainly via: (1). spectral normalization (Miyato et al., 2018) constraining weight spectral norms; (2). Jacobian regularization (Bai et al., 2021), $\mathcal{L}_{\text{jac}} = \lambda_{\text{jac}} \mathbb{E}\left[\sigma_{\max}\left(\partial \mathcal{T}_\theta / \partial \mathbf{S}\right)\right]$, directly penalizing local expansion (Definition B.17).

Lyapunov Stability. The control objective focuses on the error state $\mathcal{E}_t$, since: (i). Φ_t captures *exogenous* resource pressure beyond retrieval control; (ii). $\mathbf{H}_t$ serves as auxiliary context for prediction rather than a stabilization target. We design a quadratic Lyapunov function to quantify error "energy" (Definition D.4):

$$V(\mathcal{E}) = \mathcal{E}^\top \mathbf{P} \mathcal{E} \tag{3.4}$$

where $\mathbf{P} = \text{diag}\left(p_{\text{sem}}, p_{\text{cov}}, p_{\text{surp}}, p_{\text{rep}}\right)$ assigns weights to different error types. The key stability condition requires:

$$V(\mathcal{E}_{t+1}) \leqslant (1-\rho) \cdot V(\mathcal{E}_t) \tag{3.5}$$

for some convergence rate $\rho \in (0, 1)$. When this *Control Lyapunov Function (CLF)* condition (Definition D.3; Theorem D.2) holds, errors decay exponentially.

Theorem 3.2 (Probabilistic Ultimate Uniform Boundedness)**.** *Consider the closed-loop system subject to bounded model error $\delta_{\max}$ and stochastic system noise $\boldsymbol{\xi}_t$ with finite covariance $\boldsymbol{\Sigma}_\xi$. Under Assumptions D.4–D.6, for any confidence level $\delta \in (0, 1)$, the error energy is asymptotically bounded:*

$$\begin{aligned} &\limsup_{t \to \infty} V(\mathcal{E}_t) \leqslant b_\delta \\ &\approx \frac{2 p_{\max} \eta_{\max,\delta}^2}{\rho} \left[1 + \frac{2 p_{\max} (1-\rho)}{\rho p_{\min}}\right] \end{aligned} \tag{3.6}$$

with probability at least $1-\delta$, where $\eta_{\max,\delta} = \delta_{\max} + \sqrt{Tr(\boldsymbol{\Sigma}_\xi)/\delta}$.

This confines errors to the safety set $\Omega_\delta = \{\mathcal{E} : V(\mathcal{E}) \leqslant b_\delta\}$, preventing unbounded divergence under stochastic perturbations. Full analysis and proofs are in Appendix D.

3.4 Policy Learning

PAEC learns control policies via a three-stage pipeline.

Stage 1: Data Generation. Training trajectories from Vanilla kNN-MT are collected using five heuristic strategies with distinct "decision personalities" to ensure broad state-action coverage:

- π_{bal} (40%): Balanced three-phase reasoning (pressure $\rightarrow$ error $\rightarrow$ context);
- π_{qual} (30%): Quality-first, aggressive kNN intervention;
- π_{res} (20%): Resource-conservative, minimal intervention;
- π_{stab} (7%): Context-aware, intervening when decoder exhibits instability;
- π_{danger} (3%): Deliberate suboptimal actions as negative samples.

This mixture captures critical boundary cases. (See Appendix E for full strategy details.)

Stage 2: Lyapunov-Guided Training. The dynamics model is trained with a composite loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pred}} + w_{\text{stab}} \cdot \mathcal{L}_{\text{stab}} + \lambda_{\text{cox}} \cdot \mathcal{L}_{\text{cox}}. \quad (3.7)$$

The prediction loss $\mathcal{L}_{\text{pred}}$ uses Huber regression; $\mathcal{L}_{\text{cox}}$ provides auxiliary risk estimation for failure prediction (Definition B.14). The stability loss $\mathcal{L}_{\text{stab}}$ enforces CLF conditions (Artstein, 1983; Sontag, 1989), augmented with Control Barrier Functions and Action Dissimilarity Terms for safety and smoothness (Definitions D.5, B.15):

$$\mathcal{L}_{\text{CLF}} = \text{ReLU}\left(\sum_k w_k \Delta V_k\right), \quad (3.8)$$

$$\Delta V_k = V\left(\hat{\mathcal{E}}_{t+1}^{(k)}\right) - (1-\rho) \cdot V\left(\mathcal{E}_t\right). \quad (3.9)$$

Here, w_k are softmin weights over candidate actions, penalizing states where no action achieves Lyapunov decrease.

While eq. (3.9) defines the single-step objective, greedy optimization acts myopically. In practice, we minimize a **multi-step horizon loss** to ensure long-term stability:

$$\mathcal{L}_{N\text{-CLF}} = \sum_{i=0}^{H-1} \gamma^i \Big(\mathcal{L}_{\text{CLF}}^{(t+i)} + \lambda_{\text{CBF}} \mathcal{L}_{\text{CBF}}^{(t+i)} + \lambda_{\text{ADT}} \mathcal{L}_{\text{ADT}}^{(t+i)} \Big) - \lambda_{\text{ent}} \mathcal{H}(\pi) \quad (3.10)$$

Here, the auxiliary terms enforce safety barriers $(\mathcal{L}_{\text{CBF}})$ (Ames et al., 2019), promote temporal smoothness $(\mathcal{L}_{\text{ADT}})$, and encourage exploration (entropy $\mathcal{H}$). Proposition D.8 proves these introduce only a bounded suboptimality gap, preserving Ultimate Uniform Boundedness (UUB).

The system dynamics are unrolled over a horizon H (default $H = 3$) via **differentiable trajectory optimization**. To propagate gradients through discrete retrieval actions during training, we apply Gumbel-Softmax relaxation (Jang et al., 2017) with annealed temperature scheduling (Definition B.19). The rollout policy is reparameterized as:

$$\pi_{\text{rollout}}(\mathbf{A} \mid \mathbf{S}) = \text{Softmax}\left(\frac{-\Delta \mathbf{V}}{\tau} + \mathbf{g}\right) \quad (3.11)$$

using $\mathbf{g} = 0$ (Softmin) for deterministic smoothing or $\mathbf{g} \sim \text{Gumbel}(0, 1)$ for stochastic exploration.

Training follows a two-phase curriculum (Definition B.18; Bengio et al. 2009): Phase 1 (epochs 1–10) uses only $\mathcal{L}_{\text{pred}}$ to ensure dynamics fidelity; Phase 2 linearly increases w_{stab} to 1, aligning with the theoretical prerequisite that accurate dynamics enable tight stability control (Appendix D).

Stage 3: Implicit Policy Deployment. The trained model induces an implicit policy:

$$\pi_\theta(\mathbf{S}_t) = \arg\min_{\mathbf{A} \in \mathcal{A}_{\text{cand}}} V\left(\mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A})_{\mathcal{E}}\right) \quad (3.12)$$

where the candidate action set $\mathcal{A}_{\text{cand}}$ is generated via local perturbation of the current action (Algorithm 2).

At inference, we enumerate candidate actions, predict outcomes via $\mathcal{T}_\theta$, and select the action minimizing Lyapunov energy. This online optimization ("Online Teacher") achieves optimal stability but incurs ~ 27.7s/sentence latency.

For production deployment, we distill this into an "Offline Student" policy network π_ϕ via supervised learning (Hinton et al., 2015; Kim and Rush, 2016) on teacher demonstrations, achieving $\mathbf{93}\%$ latency reduction (~ 1.9s/sentence, see Section 4.3) while maintaining similar failure rates. Theoretically, Theorem D.5 guarantees that this student policy remains strictly stable (UUB), with the error bound relaxing only by a constant factor proportional to the distillation loss.

4 Experiments

We evaluate two PAEC variants—**Online** (gradient-based planning) and **Offline** (distilled policy)—against Vanilla and Adaptive kNN-MT on De-En translation.

4.1 Experimental Setup

Datasets. We use a mixed corpus of OPUS-100 (10%, multi-domain) and Europarl (90%, legal). Due to single-GPU memory constraints when loading three FAISS indices, we use a 50K-entry datastore for each—smaller than production scale but sufficient for framework validation. The test set comprises 2,000 stratified samples.

While this scale limitation arose from resource limits, it fortuitously creates a challenging testbed: the asymmetry between a strong NMT model and weak retrieval forces methods to either stabilize suboptimal retrieval (PAEC) or abandon it entirely (Adaptive). This "forced retrieval" regime isolates the contribution of formal stability control.

Baselines. We compare against three baselines:

(1). **Pure NMT**: Transformer-Base without kNN retrieval, representing the parametric-only approach;
(2). **Vanilla *k*NN-MT** (Khandelwal et al., 2021): fixed k=8 retrieval with uniform interpolation;
(3). **Adaptive *k*NN-MT** (Zheng et al., 2021): learned (k, λ) via Meta-k network, representing current state-of-the-art.

Implementation. We use Fairseq (Ott et al., 2019) with Transformer-Base (6 layers, 512D embeddings). Both the dynamics model $\mathcal{T}_\theta$ (Definition B.1) and offline student π_ϕ (Definition B.20) use lightweight Transformer encoders (3 layers, 64D; Table 3). The training of $\mathcal{T}_\theta$ follows curriculum learning (Definition B.18): prediction-only for epochs 1–10, then stability loss. Spectral normalization ensures Lipschitz continuity of $\mathcal{T}_\theta$ (Theorem C.2; Appendix F for ablation details).

Metrics. We report BLEU (Papineni et al., 2002), chrF, average Lyapunov error $V(\mathcal{E})$ (Definitions A.1, D.4), failure rate $(V > 0.5)$, and latency (ms/sentence).

Model	BLEU	chrF	Avg. V	Fail	Lat.
Pure NMT	36.60	62.45	**0.383**	**28.6**	**821**
Vanilla kNN	28.53	55.12	0.459	34.2	877
Adaptive	**36.68**	**62.51**	0.385	28.8	887
PAEC Online	35.44	61.28	0.396	29.3†	27732
PAEC Offline	35.22	61.05	0.397	29.3†	1941

Table 1: Performance under constrained retrieval (50K datastore, N=2,000). PAEC restricts skip-retrieval actions to test stability under forced retrieval; Adaptive is unrestricted. †Significant vs. Vanilla (McNemar's test, $p < 10^{-12}$); PAEC vs. Adaptive not significant ($p > 0.05$). Adaptive's BLEU parity with Pure NMT ($36.68 \approx 36.60$) indicates $\lambda \to 0$. Fail: failure rate (%); Lat.: latency (ms/sentence).

4.2 Main Results

Table 1 presents the overall performance on the 2,000-sample test set.

Statistical Significance. PAEC Offline reduces the failure rate from **34.2**% to **29.3**% vs. Vanilla kNN-MT (McNemar's test, $p < 10^{-12}$)—**98 fewer divergent translations**. Against Adaptive kNN-MT, the overall difference is not significant ($p > 0.05$), indicating comparable macroscopic stability; however, tail-risk analysis reveals critical differences (Section 5).

Observations. PAEC Offline exhibits a 1.4-point BLEU deficit relative to Adaptive. Near-identical metrics between Adaptive and Pure NMT (BLEU: 36.68 vs. 36.60; Fail: 28.8% vs. 28.6%) indicate the retrieval degradation ($\lambda \to 0$). PAEC maintains active retrieval, accepting a modest trade-off for formal stability guarantees. Section 5 provides attribution.

4.3 Distillation Effectiveness

A key practical concern is whether the computationally expensive Online planner can be distilled into a lightweight policy network without sacrificing stability guarantees—a question addressed theoretically by Theorem D.5 in Appendix D. Table 2 summarizes the distillation results. PAEC Offline achieves **93% latency reduction** (27.7s $\to$ 1.9s/sentence) via behavioral cloning (Definition B.21). Both variants achieve **identical failure rates** (29.30%), confirming that π_ϕ internalizes CLF-based control (Definition D.3) with average $V(\mathcal{E})$ differing by only 0.001.

Variant	Latency	Avg. V	Fail%
PAEC Online	27,732 ms	0.396	29.30
PAEC Offline	1,941 ms	0.397	29.30
Reduction	**93.0%**	+0.001	0.00

Table 2: Distillation effectiveness. PAEC Offline achieves 93% latency reduction while preserving identical failure rates.

This validates that gradient-based planning compresses into a feedforward network while preserving the teacher's risk profile. Per Theorem D.5, the distilled policy remains UUB-stable with error bound relaxed by at most $\epsilon_{\text{sub}}/\rho$.

5 Analysis

We analyze PAEC's behavior through three perspectives: performance attribution, risk-stratified evaluation, and trajectory case studies.

5.1 Attribution Analysis

The 1.4-point BLEU gap between PAEC and Adaptive kNN-MT (Table 1) stems primarily from experimental constraints rather than framework limitations.

Our setup creates asymmetry between a strong Transformer-Base (trained on millions of pairs) and a limited 50K-entry datastore. Under this configuration, Adaptive learns near-zero λ values, degenerating to Pure NMT—evidenced by their nearly identical BLEU (36.68 vs. 36.60) and failure rates (28.8% vs. 28.6%). PAEC, by contrast, enforces retrieval for trajectory stabilization even with constrained datastore quality. At production scale with billion-entry datastores, this "safety tax" should diminish as retrieval quality improves (Appendix G.1).

5.2 Risk-Stratified Evaluation

The overall metrics (Table 1) obscure a critical distinction: Adaptive's strong performance stems from *abandoning retrieval*, not *controlling it*. To isolate retrieval-dependent behavior, we stratify test samples into low-risk $\left(\text{entity coverage } \epsilon_0^{(\text{cov})}{=}0 \text{ and } V_0{<}0.8\right)$ and high-risk $\left(\epsilon_0^{(\text{cov})}{=}1 \text{ or } V_0{>}1.2\right)$ subsets, yielding 902 and 1,034 samples, respectively, after excluding 64 intermediate cases (Appendix G.2).

On high-risk samples, Pure NMT achieves Rank 1 in 45.1% of cases—skipping outperforms retrievals with a weak datastore. Vanilla kNN-MT ranks 5 in 52.1%; PAEC limits Rank 4/5 to 33.0% (vs. Vanilla's 65.8%), which exhibits stability guarantees of PAEC (Theorem 3.2) under active retrieval. Adaptive kNN-MT achieves 16.6% Rank 4/5 via $\lambda \to 0$—rational but almost bypassing retrieval. PAEC demonstrates that retrieval can be stabilized rather than abandoned.

The gap between PAEC and baselines *narrows* on high-risk samples (PAEC Offline: 30.2% Rank 1 on high-risk vs. 31.4% on low-risk), confirming PAEC's relative advantage in volatile scenarios.

5.3 Trajectory Case Studies

We examine four representative trajectories where PAEC significantly outperforms Adaptive kNN-MT (Appendix G.3), revealing three patterns:

1. **Variable Divergence Timing:** Separation occurs at different stages—early (steps 5–8) or late (steps 15–20). The dynamics model $\mathcal{T}_\theta$ (Definition B.1) identifies divergence risk via multi-step lookahead *before* it manifests in immediate metrics.
2. **Adaptive Degenerates to Pure NMT:** Adaptive trajectories closely track Pure NMT in all cases, confirming that the Meta-k network predicts near-zero λ and abandons retrieval.
3. **Lookahead Enables Preemptive Intervention:** At critical decision points where confidence appears adequate, PAEC's trajectory prediction triggers retrieval while Adaptive abstains—the short-term cost is outweighed by long-term error reduction.

These trajectories reveal distinct strategies for handling suboptimal retrieval. Adaptive kNN-MT correctly identifies limited datastore value and learns $\lambda \to 0$, achieving robustness by *bypassing* retrieval. PAEC, constrained to maintain retrieval, achieves robustness by *controlling* retrieval—demonstrating stability guarantees under suboptimal datastores where the other kNN methods learn to disengage. The observed ΔV gains in Figure 6 quantify this extracted value. The experimental design—constraining PAEC to evaluate retrieval control rather than retrieval avoidance—is crucial to validating the Lyapunov framework's contribution in kNN algorithm: stability guarantees matter when retrieval cannot be abandoned.

6 Related Work

***k*NN-MT and Adaptive Retrieval.** kNN-MT (Khandelwal et al., 2021) enables non-parametric domain adaptation via external datastore retrieval. While subsequent research optimizes the retrieval *mechanism*—through Adaptive kNN-MT (Zheng et al., 2021), token-level filtering acceleration (Shi et al., 2024), or INK representation enhancement (Zhu et al., 2023)–PAEC regulates the underlying **decision dynamics**. By framing retrieval as a dynamical control problem, PAEC provides an orthogonal stability-guaranteeing layer compatible with diverse kNN-MT architectures.

Exposure Bias in Autoregressive Generation. Exposure bias (Ranzato et al., 2016) stems from the train-test distribution mismatch, often inducing hallucinations and domain shifts in NMT (Wang and Sennrich, 2020). Unlike prior works that attempt to eliminate the bias at the source, PAEC adopts an orthogonal strategy: we utilize Lyapunov stability constraints to bound its consequences, theoretically guaranteeing error convergence regardless of decoding trajectory length.

Control Theory in Machine Learning. Control-theoretic perspectives have enriched deep learning: Model Predictive Control (MPC) (Mayne et al., 2000) for sequential decision-making; Control Barrier Functions (CBF) (Ames et al., 2019) for safety guarantees; and neural Lyapunov functions (Wu et al., 2023) for learning stabilizing controllers. We adapt these tools to the NLP domain, designing a Lyapunov function over translation error states and enforcing CLF conditions (Artstein, 1983; Sontag, 1989) during dynamics model training.

Lipschitz Constraints in Neural Networks. Lipschitz continuity ensures bounded sensitivity to input perturbations. Spectral normalization (Miyato et al., 2018) constrains weight matrices to unit spectral norm; analytical bounds (Scaman and Virmaux, 2018; Avant and Morgansen, 2024) characterize network-wide constants. We employ spectral normalization to guarantee our dynamics model $\mathcal{T}_\theta$ is Lipschitz continuous (Theorem C.2), a prerequisite for the stability proofs in Appendix D.

7 Conclusion

We presented PAEC, a control-theoretic framework for stabilizing kNN-MT retrieval under suboptimal datastore conditions. While adaptive methods learn to abandon retrieval when datastore quality is limited, PAEC reformulates retrieval control as a dynamical system stabilization problem. By modeling decoding as an 11-dimensional state evolution with a Lipschitz-continuous dynamics model $\mathcal{T}_\theta$ (Theorem C.2) and enforcing Control Lyapunov Function conditions, PAEC achieves Ultimate Uniform Boundedness (Theorem 3.2), guaranteeing bounded error accumulation while maintaining active retrieval.

In proof-of-concept experiments under forced retrieval (OPUS-100/Europarl, 50K-entry datastore), PAEC reduces failure rate from 34.2% to 29.3% vs. Vanilla kNN-MT $\left(p < 10^{-12}\right)$. Under these conditions, Adaptive kNN-MT achieves strong overall metrics by degenerating to Pure NMT, whereas PAEC maintains active retrieval—demonstrating that formal stability guarantees enable tail-risk control without abandoning the retrieval mechanism.

As a control layer orthogonal to the underlying retrieval mechanism, PAEC is designed to integrate with advances in retrieval efficiency. Policy distillation achieves 93% latency reduction while preserving stability guarantees (Theorem D.5).

Future Work. Promising directions include:

(1). **Scale validation**: Evaluating PAEC with larger datastores (1M–1B entries) to test our hypothesis that the BLEU trade-off diminishes with improved retrieval quality;
(2). **Language coverage**: Extending experiments to distant language pairs and low-resource settings;
(3). **Neural Lyapunov functions**: Learning $V(\mathcal{E})$ end-to-end rather than using fixed quadratic forms;
(4). **Integration with retrieval advances**: Empirically validating the orthogonality of PAEC with token-level filtering (Shi et al., 2024) and representation smoothing (Zhu et al., 2023);
(5). **Task generalization**: Extending to other autoregressive tasks such as dialogue generation and code synthesis.

8 Limitations

Experimental Scale. Due to memory constraints when initializing multiple FAISS indices on a single GPU (NVIDIA A100), our experiments use a 50K-entry datastore—orders of magnitude smaller than production systems. This creates asymmetry between a strong parametric model and weak retrieval components, amplifying the observed BLEU trade-off (Section 4.2). The theoretical framework is scale-agnostic; larger-scale validation is important future work.

Language Coverage. Experiments are confined to German-English; multilingual generalization, especially for distant language pairs (e.g., En-Zh, En-Ja), remains unverified.

Lyapunov Function Design. The quadratic Lyapunov function with a diagonal weight matrix $\mathbf{P}$ assumes that error components are independent, which may not capture cross-component interactions. Neural Lyapunov functions learned end-to-end represent a promising direction. See Appendix G.4 for comprehensive discussions.

9 Ethical Considerations

This work adheres to the ACL Code of Ethics. AI writing assistants (Claude) were used for English language polishing of the paper and debugging assistance for code implementations; all intellectual contributions are the author's original work. Potential ethical impacts include:

(1). Biases inherited from NMT training data may propagate through translation;
(2). The stability guarantees bound error accumulation but do not eliminate inherent data biases;
(3). Improved MT systems could potentially be misused for generating misleading translations at scale.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. This work was conducted independently without institutional funding or dedicated computational infrastructure; all experiments were performed on a single NVIDIA A100 GPU accessed through cloud computing services. We hope this resource context helps in the interpretation of our experimental scope.

References

Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. 2019. Control barrier functions: Theory and applications. In *18th European Control Conference*, Naples, Italy.

Zvi Artstein. 1983. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173.

Trevor Avant and Kristi A. Morgansen. 2024. Analytical bounds on the local lipschitz constants of relu networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13902–13913.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. `https://arxiv.org/abs/1607.06450`. *Preprint*, arXiv:1607.06450.

Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. 2021. Stabilizing equilibrium models by jacobian regularization. In *Proceedings of the 38th International Conference on Machine Learning*, Virtual Event.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Quebec, Canada.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland.

Randy A. Freeman and Petar V. Kokotovic. 1996. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Birkhäuser Boston.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*, Montreal, Canada.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*.

Hassan K. Khalil. 2002. *Nonlinear Systems*, 3 edition. Prentice Hall, Upper Saddle River, NJ, USA.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *9th International Conference on Learning Representations*, Virtual Event. OpenReview.net.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA.

Aleksandr Mikhailovich Lyapunov. 1992. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. 2000. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations*, Vancouver, BC, Canada.

Pravin Nair. 2025. Softmax is 1/2-lipschitz: A tight bound across all ℓ_p norms. `https://arxiv.org/abs/2510.23012`. *Preprint*, arXiv:2510.23012. Under review; version v1.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations*, San Juan, Puerto Rico.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China.

R. Tyrrell Rockafellar and Stanislav Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41.

Kevin Scaman and Aladin Virmaux. 2018. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems 31*, page 3835–3844, Montreal, Canada. Neural Information Processing Systems.

Xiangyu Shi, Yunlong Liang, Jinan Xu, and Yufeng Chen. 2024. Towards faster k-nearest-neighbor machine translation. *Advances in Artificial Intelligence and Machine Learning*, 4(1):1943–1958.

Eduardo D. Sontag. 1989. A 'universal' construction of artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117–123.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online.

Junlin Wu, Andrew Clark, Yiannis Kantaros, and Yevgeniy Vorobeychik. 2023. Neural lyapunov control for discrete-time systems. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA, USA.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online. Association for Computational Linguistics.

Wenhao Zhu, Jingjing Xu, Shujian Huang, Lingpeng Kong, and Jiajun Chen. 2023. Ink: Injecting knn knowledge in nearest neighbor machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada.

A State Space: Definitions and Proofs

This appendix formalizes all state space components and establishes their mathematical properties.

A.1 Error State Vector $\mathcal{E}_t$

Definition A.1 (Error State Vector)**.** *The error state $\mathcal{E}_t \in \mathbb{R}^4$ quantifies the deviation in translation quality:*

$$\mathcal{E}_t = \begin{bmatrix} \epsilon_t^{(sem)} \\ \epsilon_t^{(cov)} \\ \epsilon_t^{(surp)} \\ \epsilon_t^{(rep)} \end{bmatrix}. \tag{A.1}$$

A.1.1 Semantic Drift Error $\epsilon_t^{(\text{sem})}$

Definition A.2 (Semantic Drift)**.** *Let $\mathbf{e}_{src}, \mathbf{e}_{hyp}^{(t)} \in \mathbb{R}^d$ denote the sentence embeddings of the source text and the generated prefix at step t, respectively. The semantic drift is:*

$$\epsilon_t^{(sem)} = 1 - \frac{\mathbf{e}_{src} \cdot \mathbf{e}_{hyp}^{(t)}}{\|\mathbf{e}_{src}\|_2 \left\|\mathbf{e}_{hyp}^{(t)}\right\|_2} \in [0, 2]\,. \tag{A.2}$$

The sentence embeddings are computed via pre-trained multilingual encoders such as LaBSE (Feng et al., 2022) (by default) or Sentence-BERT (Reimers and Gurevych, 2019).

Proposition A.1 (Properties of Semantic Drift)**.** *The semantic drift $\epsilon_t^{(sem)}$ satisfies:*

(i) ***Range:*** *$\epsilon_t^{(sem)} \in [0, 2]$.*

(ii) ***Continuity:*** *$\epsilon_t^{(sem)}$ is continuous w.r.t. the embedding vectors on $\left(\mathbb{R}^d \setminus \{\mathbf{0}\}\right)^2$.*

Proof. (i) **Range:** By the Cauchy-Schwarz inequality, for any non-zero vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, the cosine similarity satisfies $\cos\theta = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|_2\|\mathbf{v}\|_2} \in [-1, 1]$. Therefore, $\epsilon_t^{(\text{sem})} = 1 - \cos\theta \in [0, 2]$, with the bounds achieved when $\mathbf{e}_{\text{src}}$ and $\mathbf{e}_{\text{hyp}}^{(t)}$ are parallel ($\cos\theta = 1$) or anti-parallel ($\cos\theta = -1$), respectively.

(ii) **Continuity:** The function $f(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|_2\|\mathbf{v}\|_2}$ is continuous on $\left(\mathbb{R}^d \setminus \{\mathbf{0}\}\right)^2$ as a composition of continuous operations (dot product, norm, division by non-zero denominator, and subtraction).

□

A.1.2 Entity Coverage Error $\epsilon_t^{(\text{cov})}$

Definition A.3 (Entity Coverage Error)**.** *Let $\mathcal{N}_{src}$ be the set of named entities in the source sentence, and $\mathcal{N}_{covered}^{(t)}$ be the entities successfully translated by step t:*

$$\epsilon_t^{(cov)} = 1 - \left|\mathcal{N}_{covered}^{(t)}\right| / \left|\mathcal{N}_{src}\right| \in [0, 1]\,, \tag{A.3}$$

with the convention $\epsilon_t^{(cov)} = 0$ when $\mathcal{N}_{src} = \emptyset$.

To determine $\mathcal{N}_{\text{covered}}^{(t)}$ in eq. (A.3), we employ a two-stage entity alignment procedure described in Algorithm 1.

Algorithm 1 Entity Alignment for Coverage

Require: Source entities $\mathcal{N}_{\text{src}}$, generated prefix $y_{<t}$, similarity threshold τ
Ensure: Covered entity set $\mathcal{N}_{\text{covered}}^{(t)}$

$\mathcal{N}_{\text{covered}}^{(t)} \leftarrow \emptyset$
$\mathcal{N}_{\text{hyp}} \leftarrow \text{NER}\,(y_{<t})$ ▷ Extract entities from hypothesis
$\mathcal{N}_{\text{src}}^{\text{rem}} \leftarrow \mathcal{N}_{\text{src}}$; $\mathcal{N}_{\text{hyp}}^{\text{rem}} \leftarrow \mathcal{N}_{\text{hyp}}$
Phase 1: Exact Matching
for each $e_s \in \mathcal{N}_{\text{src}}^{\text{rem}}$ **do**
  for each $e_h \in \mathcal{N}_{\text{hyp}}^{\text{rem}}$ **do**
    if lower (e_s) = lower (e_h) **then**
      $\mathcal{N}_{\text{covered}}^{(t)} \leftarrow \mathcal{N}_{\text{covered}}^{(t)} \cup \{e_s\}$
      $\mathcal{N}_{\text{src}}^{\text{rem}} \leftarrow \mathcal{N}_{\text{src}}^{\text{rem}} \setminus \{e_s\}$; $\mathcal{N}_{\text{hyp}}^{\text{rem}} \leftarrow \mathcal{N}_{\text{hyp}}^{\text{rem}} \setminus \{e_h\}$
      break
    end if
  end for
end for
Phase 2: Semantic Matching via the Hungarian Algorithm
Construct bipartite $G = \left(\mathcal{N}_{\text{src}}^{\text{rem}}, \mathcal{N}_{\text{hyp}}^{\text{rem}}, E\right)$
for each $e_s \in \mathcal{N}_{\text{src}}^{\text{rem}}$, $e_h \in \mathcal{N}_{\text{hyp}}^{\text{rem}}$ **do**
  sim $\leftarrow \cos\left(\text{Emb}\,(e_s)\,, \text{Emb}\,(e_h)\right)$
  if sim $\geqslant \tau$ **then**
    Add edge (e_s, e_h) with sim to E
  end if
end for
$\mathcal{M}^* \leftarrow \text{Hungarian}\,(G)$ ▷ Maximum weight bipartite matching
for each $(e_s, e_h) \in \mathcal{M}^*$ **do**
  $\mathcal{N}_{\text{covered}}^{(t)} \leftarrow \mathcal{N}_{\text{covered}}^{(t)} \cup \{e_s\}$
end for
return $\mathcal{N}_{\text{covered}}^{(t)}$

The two-phase design balances efficiency and robustness: exact matching handles preserved entities while semantic matching via the Hungarian algorithm captures cross-lingual correspondences (e.g., "United States" ↔ "Vereinigten Staaten"). The threshold $\tau = 0.75$ balances precision and recall.

A.1.3 Uncertainty Error $\epsilon_t^{(\text{surp})}$

Definition A.4 (Prediction Uncertainty/Surprisal)**.** *Let $P_{LM}(\cdot \mid y_{<t})$ denote the probability distribution over the vocabulary $\mathcal{V}$ at step t. The uncertainty error is defined as the normalized Shannon entropy of the predictive distribution:*

$$\epsilon_t^{(surp)} = \frac{H(P_{LM})}{\log_2 |\mathcal{V}|} = \frac{-\sum_{w \in \mathcal{V}} p(w \mid y_{<t}) \log_2 p(w \mid y_{<t})}{\log_2 |\mathcal{V}|}, \quad \text{(A.4)}$$

where $|\mathcal{V}|$ is the vocabulary size.

This metric quantifies model uncertainty independent of the sampled token.

A.1.4 Repetition Error $\epsilon_t^{(\text{rep})}$

Definition A.5 (Semantic Self-Similarity)**.** *Let $\mathbf{H}_L \in \mathbb{R}^{L \times d}$ be the matrix of normalized token embeddings for the last L generated tokens where L denotes the effective window size at step t. We compute the pairwise cosine similarity matrix $\mathbf{S} = \mathbf{H}_L \mathbf{H}_L^\top$. The repetition error is the mean of the upper triangular off-diagonal elements:*

$$\epsilon_t^{(rep)} = \frac{2}{L(L-1)} \sum_{1 \leqslant i < j \leqslant L} \mathbf{S}_{ij}. \quad \text{(A.5)}$$

This metric captures redundancy in the continuous semantic space, detecting looping behaviors even when surface forms differ slightly.

Proposition A.2 (Topological Properties of Error State)**.** *Let $\mathcal{S}_\mathcal{E} \subset \mathbb{R}^4$ be the state space of $\mathcal{E}_t$. The space satisfies the following topological properties:*

(a) ***Compactness:*** *$\mathcal{S}_\mathcal{E}$ is a compact set.*
(b) ***Convexity:*** *$\mathcal{S}_\mathcal{E}$ is a convex set.*
(c) ***Boundedness:*** *For any $\mathcal{E}_t \in \mathcal{S}_\mathcal{E}$, $\|\mathcal{E}_t\|_2 \leqslant \sqrt{7}$.*

Proof. **(a). Compactness:** Each component has a closed bounded domain: $\epsilon^{(\text{sem})} \in [0, 2]$, others in $[0, 1]$. Thus $\mathcal{S}_\mathcal{E} = [0, 2] \times [0, 1]^3$ is compact by the Heine-Borel theorem. **(b). Convexity:** As a hyperrectangle (Cartesian product of intervals), $\mathcal{S}_\mathcal{E}$ is trivially convex. For any $\mathbf{x}, \mathbf{y} \in \mathcal{S}_\mathcal{E}$ and $\theta \in [0, 1]$, $\theta\mathbf{x} + (1-\theta)\mathbf{y}$ remains within the bounds defined by the intervals. **(c). Boundedness:** The Euclidean norm is maximized at the vertex $(2, 1, 1, 1)$:

$$\|\mathcal{E}_t\|_2 \leqslant \sqrt{2^2 + 1^2 + 1^2 + 1^2} = \sqrt{7}.$$

□

A.2 Pressure State Vector Φ_t

The pressure state $\Phi_t \in \mathbb{R}^3$ is a differentiable measure of the system's physical constraints. We apply strict normalization to ensure numerical stability and to bound the state space for Lyapunov analysis (Appendix D).

Definition A.6 (Pressure State Components)**.** *The pressure state $\Phi_t = \left[\phi_t^{(lat)}, \phi_t^{(mem)}, \phi_t^{(thr)}\right]^\top$ is computed using the sigmoid function $\sigma(z) = (1 + e^{-z})^{-1}$ and clipped to $[\epsilon, 1 - \epsilon]$ (where $\epsilon = 10^{-6}$):*

(i) ***Latency Pressure*** *$\phi_t^{(lat)}$: Penalizes both high latency and high jitter (volatility).*

$$\phi_t^{(lat)} = \sigma\left(w_1 \frac{L_t}{L_{SLA}} + w_2 \frac{|\Delta L_t|}{L_{SLA}}\right),$$

where L_t is the moving average latency, $|\Delta L_t|$ is the absolute difference between consecutive latency measurements capturing jitter, and L_{SLA} is the Service-Level-Agreement latency threshold (e.g., 100 ms).

(ii) ***Memory Pressure*** *$\phi_t^{(mem)}$: Predicts out-of-memory (OOM) risks based on usage and growth rate.*

$$\phi_t^{(mem)} = \sigma\left(w_3 \frac{M_t}{M_{total}} + w_4 \frac{\Delta M_t}{M_{total}}\right),$$

where M_t is current usage, M_{total} is total capacity, and $\Delta M_t = M_t - M_{t-1}$.

(iii) ***Throughput Pressure*** *$\phi_t^{(thr)}$: Defined by an asymmetric function of the throughput gap ratio $r_t = (R_{opt} - R_t)/R_{opt}$.*

$$\phi_t^{(thr)} = \begin{cases} \sigma\left(w_{5,def} \cdot |r_t| - w_6\right) \\ \quad \text{if } r_t > 0 \ (\textit{Deficit}); \\ \sigma\left(w_{5,sur} \cdot r_t - w_6\right) \\ \quad \text{if } r_t \leqslant 0 \ (\textit{Surplus}). \end{cases}$$

Here, $w_{5,def} \gg w_{5,sur}$ ensures the system reacts aggressively to throughput drops but relaxes slowly when capacity is recovered.

Proposition A.3 (Topological Properties of Pressure State Space)**.** *The pressure state space $\mathcal{S}_\Phi \subset \mathbb{R}^3$ satisfies:*

(a) ***Compactness:*** *$\mathcal{S}_\Phi \subset [\epsilon, 1-\epsilon]^3$ is compact.*
(b) ***Lipschitz Continuity:*** *The mapping from physical metrics to Φ_t is globally Lipschitz continuous.*

Proof. (a) **Compactness:** Values are clipped to $[\epsilon, 1-\epsilon]$. As a closed bounded subset of $\mathbb{R}^3$, $\mathcal{S}_\Phi$ is compact by the Heine-Borel theorem.

(b) **Lipschitz Continuity:** Each pressure component has the form $\phi = \sigma \circ g$. Since $\sigma'(z) \leqslant 0.25$ for all z, it suffices to show each inner function g is Lipschitz.
Latency: g_{lat} is a linear combination of L_t and $|L_t - L_{t-1}|$. Since the absolute value function is 1-Lipschitz and linear combinations of Lipschitz functions are Lipschitz, g_{lat} is Lipschitz.
Memory: g_{mem} is a linear combination of M_t and M_{t-1} with bounded coefficients, hence Lipschitz.
Throughput: g_{thr} is continuous and piecewise linear with slopes $w_{5,\text{def}}$ $(r_t > 0)$ and $w_{5,\text{sur}}$ $(r_t \leqslant 0)$. Hence $\text{Lip}(g_{\text{thr}}) = \max(w_{5,\text{def}}, w_{5,\text{sur}})$.
Thus, Φ_t is globally Lipschitz continuous.

□

A.3 Generative Context State Vector $\mathbf{H}_t$

The context state $\mathbf{H}_t \in \mathbb{R}^4$ captures the internal cognitive dynamics of the NMT decoder, encompassing attention distribution, representation stability, and predictive uncertainty.

Definition A.7 (Context State Components)**.** *Let $\boldsymbol{\alpha}_t \in \Delta^{T_s-1}$ be the cross-attention weights, $\mathbf{q}_t$ the current decoder query, $\mathcal{Q}_{<t} = \{\mathbf{q}_0, \ldots, \mathbf{q}_{t-1}\}$ the history of past queries, and $\mathbf{z}_t \in \mathbb{R}^{|V|}$ the output logits. The components of $\mathbf{H}_t = [H_t^{(foc)}, H_t^{(con)}, H_t^{(stab)}, H_t^{(vol)}]^\top$ are defined as:*

(i) ***Focus (Attention Faithfulness)*** *$H_t^{(foc)}$: The cumulative attention mass allocated to* uncovered *named entities $\mathcal{I}_{NE}$:*

$$H_t^{(foc)} = \sum_{j \in \mathcal{I}_{NE}} \alpha_{t,j}.$$

(ii) ***Consistency*** *$H_t^{(con)}$: Let $\mathbf{h}_j^{enc}$ be the hidden states of the encoder. Given the alignment between the decoding query and the retrieved source context vector $\mathbf{c}_t = \sum_{j=1}^{T_s} \alpha_{t,j} \mathbf{h}_j^{enc}$, the metric $H_t^{(con)}$ is defined as:*

$$H_t^{(con)} = \frac{1}{2}\left(1 + \frac{\mathbf{q}_t \cdot \mathbf{c}_t}{\|\mathbf{q}_t\| \, \|\mathbf{c}_t\|}\right).$$

(iii) ***Stability*** *$H_t^{(stab)}$: The trajectory coherence, measured by the normalized mean cosine similarity between the current query and the entire query history $\mathcal{Q}_{<t}$. For $t \geqslant 1$:*

$$H_t^{(stab)} = \frac{1}{2}\left(1 + \frac{1}{|\mathcal{Q}_{<t}|} \sum_{\mathbf{q}_k \in \mathcal{Q}_{<t}} \frac{\mathbf{q}_t \cdot \mathbf{q}_k}{\|\mathbf{q}_t\| \, \|\mathbf{q}_k\|}\right).$$

For the initial step $t = 0$, we define $H_0^{(stab)} = 0$ as no historical context is available.

(iv) ***Volatility*** *$H_t^{(vol)}$: The temporal variability of model confidence. Let confidence c_t be derived from the margin between the top logit $z_t^{(1)}$ and the average of the remaining top-k logits $\left(z_t^{(2\ldots k)}\right)$:*

$$c_t = \sigma\left\{\frac{1}{2}\left[z_t^{(1)} - \frac{1}{k-1} \cdot \sum_{j=2}^{k} z_t^{(j)}\right]\right\},$$

where $\sigma(\cdot)$ denotes the Sigmoid function. The volatility is defined as the smoothed population standard deviation over the confidence history $\mathcal{C}_{hist}$ of length T, utilizing a regularization constant $\eta = 10^{-6}$:

$$H_t^{(vol)} = \sqrt{\frac{1}{T}\sum_{\tau=1}^{T}(c_\tau - \bar{c})^2 + \eta}.$$

Proposition A.4 (Topological Properties of Context State)**.** *The context state space $\mathcal{S}_{\mathbf{H}}$ satisfies the following strict properties:*

(a) ***Compactness:*** *The state space is a compact subset of the unit hypercube, $\mathcal{S}_{\mathbf{H}} \subset [0,1]^4$.*
(b) ***Lipschitz Continuity:*** *The function generating $\mathbf{H}_t$ is Lipschitz continuous on the bounded domain of hidden states.*

Proof. **(a). Compactness and Boundedness:**

- $H_t^{(\text{foc})}$: partial sum of $\boldsymbol{\alpha}_t \in \Delta^{T_s-1}$, bounded in $[0,1]$.

- $H_t^{(\text{con})}, H_t^{(\text{stab})}$: derived from cosine similarities via $\frac{1}{2}(1+x)$, mapping $[-1,1] \to [0,1]$.
- $H_t^{(\text{vol})}$: standard deviation of $c_\tau \in (0,1)$. By Popoviciu's inequality, the variance satisfies $\mathbb{V}[c] \leqslant 0.25$, yielding $H_t^{(\text{vol})} = \sqrt{\mathbb{V}[c] + \eta} \leqslant \sqrt{0.25 + \eta} \approx 0.5 \in [0,1]$.

Hence, as the Cartesian product of closed and bounded intervals, $\mathcal{S}_{\mathbf{H}}$ is compact by the Heine-Borel theorem.

(b). Lipschitz Continuity: We bound the Lipschitz constant of each component w.r.t. the decoder hidden state $\mathbf{h}_t$:

- **Focus** $H_t^{(\text{foc})}$: As the partial sum of the Softmax outputs for cross-attention weights $\boldsymbol{\alpha}_t$, since Softmax is $1/2$-Lipschitz w.r.t. the ℓ_2 norm (Nair, 2025), $L_{\text{foc}} \leqslant 1/2$.
- **Consistency/Stability** $H_t^{(\text{con})}, H_t^{(\text{stab})}$: We assume that post-LayerNorm hidden states satisfy $\|\mathbf{h}\| \geqslant \epsilon_{\text{LN}} > 0$, which holds in practice as LayerNorm (Ba et al., 2016) prevents degenerate zero-norm representations. Under this assumption, the gradient of cosine similarity is bounded by $\frac{2}{\epsilon_{\text{LN}}}$. By $\frac{1}{2}(1+\cos\theta)$, the Lipschitz constant is $L_{\text{con}}, L_{\text{stab}} \leqslant \frac{1}{\epsilon_{\text{LN}}}$.
- **Volatility** $H_t^{(\text{vol})}$: Let $v = \text{Var}[\mathbf{c}]$ represent the empirical variance of the confidence sequence $\mathbf{c} = (c_1, \ldots, c_T)$. The mapping $f(v) = \sqrt{v+\eta}$ is differentiable with $f'(v) = 1/(2\sqrt{v+\eta})$. Given $v \geqslant 0$, the derivative is globally bounded by $L_f \leqslant 1/(2\sqrt{\eta})$.

  For the variance function $v(\mathbf{c}) = \frac{1}{T}\sum_\tau (c_\tau - \bar{c})^2$, the gradient satisfies $\|\nabla_{\mathbf{c}} v\|_2 = \frac{2\sqrt{v}}{\sqrt{T}} \leqslant \frac{1}{\sqrt{T}}$ by Popoviciu's inequality ($v \leqslant 1/4$). Since each $c_\tau = \sigma(m_\tau)$ with $\sigma' \leqslant 0.25$, the chain rule yields $L_{\text{var}} \leqslant \frac{1}{4\sqrt{T}}$. Composing with f, we obtain $L_{\text{vol}} \leqslant L_{\text{var}} \cdot L_f = \frac{1}{4\sqrt{T}} \cdot \frac{1}{2\sqrt{\eta}} < \infty$.

By part (a), the state space $\mathcal{S}_{\mathbf{H}} \subset [0,1]^4$ is compact. On this compact domain, each component mapping is Lipschitz continuous with a finite constant. By the composition lemma for Lipschitz functions, the composite mapping to $\mathbf{H}_t$ has a Lipschitz constant $L_{\mathbf{H}} = \sqrt{L_{\text{foc}}^2 + L_{\text{con}}^2 + L_{\text{stab}}^2 + L_{\text{vol}}^2} < \infty$. □

A.4 Standardization Transformation

Definition A.8 (Hybrid Standardization)**.** *The transformation $\Psi : \mathbb{R}^{11} \to \mathbb{R}^{11}$ applies distinct scaling strategies based on the distributional properties of state components:*

- ***Affine Standardization:*** *For components exhibiting unimodal or approximate Gaussian distributions, we apply $\Psi_{std}(x) = (x-\mu)/\sigma$ to $\mathcal{E}_t \setminus \epsilon_t^{(cov)}$ and $\mathbf{H}_t \setminus H_t^{(foc)}$.*
- ***Quantile Normalization:*** *For components with complex, multi-modal distributions or bounded intervals, we apply $\Psi_{quant}(x) = \Phi^{-1}\left(\tilde{F}(x)\right)$, where $\tilde{F}$ is a smoothed empirical CDF via linear interpolation and Φ^{-1} is the standard normal inverse CDF; we applied Ψ_{quant} on $\epsilon_t^{(cov)}$, full Φ_t, and $H_t^{(foc)}$ in our implementation.*

Proposition A.5 (Properties of Standardization)**.** *The hybrid standardization Ψ satisfies:*

(i) ***Bijectivity:*** *Ψ is invertible on the effective support with a continuous inverse.*
(ii) ***Lipschitz Continuity:*** *Ψ is Lipschitz continuous on the compact state space.*
(iii) ***Normalization:*** *Ψ_{std} outputs have zero mean and unit variance; Ψ_{quant} outputs follow $\mathcal{N}(0,1)$ on the training support.*

Proof. (i) **Bijectivity:** Ψ_{std} is affine with $\sigma > 0$, hence bijective. Ψ_{quant} uses a smoothed CDF $\tilde{F}$ via linear interpolation, which is strictly increasing and thus invertible on the interpolation domain. Boundary clipping to $(\varepsilon_{\text{clip}}, 1-\varepsilon_{\text{clip}})$ before applying Φ^{-1} ensures finite outputs; the inverse is well-defined on the range $\left(\Phi^{-1}(\varepsilon_{\text{clip}}), \Phi^{-1}(1-\varepsilon_{\text{clip}})\right)$.

(ii) **Lipschitz Continuity:** For Ψ_{std}: Affine with $L_{\text{std}}^{(j)} = 1/\sigma_j$. For Ψ_{quant}: Piecewise linear with strict monotonicity enforced via tie-breaking perturbation $\left(x_{i+1} \leftarrow \max(x_{i+1}, x_i + \varepsilon_{\text{tie}})\right)$. Each segment slope is bounded by $\Delta y_{\max}/\varepsilon_{\text{tie}} < \infty$. On the compact domain (Propositions A.2–A.4), we have composite $L_\Psi = \max_j L^{(j)} < \infty$, where $L_\Psi \approx 126$ empirically (Remark C.3).

(iii) **Normalization:** Ψ_{std} centers and scales to zero mean and unit variance (distribution shape preserved). Ψ_{quant} maps any continuous distribution to $\mathcal{N}(0,1)$ via the probability integral transform. □

B Dynamics Model Architecture Details

This appendix provides complete specifications of the Transformer-based dynamics model $\mathcal{T}_\theta$.

Definition B.1 (Dynamics Model $\mathcal{T}_\theta$)**.** *The dynamics model $\mathcal{T}_\theta : \mathbb{R}^{11} \times \mathbb{R}^6 \times \mathcal{H} \to \mathbb{R}^8$ predicts the next internal state components:*

$$\left[\hat{\mathcal{E}}_{t+1}, \hat{\mathbf{H}}_{t+1}\right] = \mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}_t; \mathbf{S}_{hist}, \mathbf{A}_{hist}\right), \quad \text{(B.1)}$$

*where $\mathcal{H} = \mathbb{R}^{w\times 11} \times \mathbb{R}^{w\times 6}$ denotes the history space. The full next state is assembled alongside the **Inertia Assumption** of Φ_t ($\Phi_{t+1} \approx \Phi_t$):*

$$\hat{\mathbf{S}}_{t+1} = \left[\hat{\mathcal{E}}_{t+1}; \Phi_t; \hat{\mathbf{H}}_{t+1}\right] \in \mathbb{R}^{11}. \quad \text{(B.2)}$$

The model is composed of:

1. ***Embedding Layers** (Definition B.4): Project states and actions to d-dimensional space with spectral normalization.*
2. ***Context Encoding** (Definition B.6): Form sequence, add positional encoding, and extract temporal context $\mathbf{z}_t$ via Transformer Encoder (Definitions B.7, B.8).*
3. ***Action Fusion** (Definition B.9): Fuse context $\mathbf{z}_t$ with action embedding $\mathbf{a}_{emb}$ via cross-attention or concatenation.*
4. ***Prediction Heads** (Definition B.10): Output predicted $\hat{\mathcal{E}}_{t+1}$, $\hat{\mathbf{H}}_{t+1}$, and auxiliary Cox risk score.*

The model is trained with a composite loss (Definition B.16) combining prediction accuracy ($\mathcal{L}_{pred}$) and stability constraints ($\mathcal{L}_{stab}$).

B.1 Input Representation

Definition B.2 (Complete Input)**.** *The dynamics model receives:*

$$\mathbf{X} = \left[\mathbf{S}_t; \mathbf{S}_{hist}; \mathbf{A}_t; \mathbf{A}_{hist}; \mathbf{I}_{aux}\right], \quad \text{(B.3)}$$

where:

- $\mathbf{S}_t \in \mathbb{R}^{11}$*: Current state (standardized);*
- $\mathbf{S}_{hist} \in \mathbb{R}^{w\times 11}$*: Historical states over the window w;*
- $\mathbf{A}_t \in \mathbb{R}^6$*: Current action $\big($4D one-hot index type $\in \{$None, Exact (`IndexFlatL2`), HNSW, IVFPQ$\}$, 2D continuous $k, \lambda\big)$;*
- $\mathbf{A}_{hist} \in \mathbb{R}^{w\times 6}$*: Historical actions;*
- $\mathbf{I}_{aux}$*: Optional auxiliary inputs (e.g., source embeddings, decoder hidden states).*

B.2 Action Space

Definition B.3 (Action Vector)**.** *The action $\mathbf{A}_t \in \mathbb{R}^6$ consists of:*

$$\mathbf{A}_t = \Bigg[\underbrace{a_{none}, a_{exact}, a_{hnsw}, a_{ivf\_pq}}_{\substack{\text{one-hot}\\ \text{index type}}}, \underbrace{\tilde{k}, \tilde{\lambda}}_{\substack{\text{normalized}\\ \text{continuous}}}\Bigg], \quad \text{(B.4)}$$

where $\tilde{k} = k/k_{\max} \in [0, 1]$ and $\tilde{\lambda} \in [0, 1]$ are the interpolation weights.

Algorithm 2 Local Candidate Action Generation

Require: Current Teacher Action $\mathbf{A}_t \in \mathbb{R}^6$, perturbation magnitude δ (default 0.25), active dimensions N_{dir}
Ensure: Candidate Action Set $\mathcal{A}_{\text{cand}}$

$\mathcal{A}_{\text{cand}} \leftarrow \{\mathbf{A}_t\}$ ▷ Include teacher action
Let $\mathbf{A}_{\text{disc}} = \mathbf{A}_t[0:4]$ ▷ Fixed one-hot part
Let $\mathbf{A}_{\text{cont}} = \mathbf{A}_t[4:6]$ ▷ Continuous (k, λ)
for $i \leftarrow 0$ to $N_{\text{dir}} - 1$ **do** ▷ Iterate over continuous dimensions to perturb
  $\boldsymbol{\Delta} \leftarrow \mathbf{0} \in \mathbb{R}^2$
  $\boldsymbol{\Delta}[i] \leftarrow \delta$
  ▷ Generate +/- perturbations
  $\mathbf{A}_{\text{plus}} \leftarrow [\mathbf{A}_{\text{disc}}; \text{clip}(\mathbf{A}_{\text{cont}} + \boldsymbol{\Delta}, 0, 1)]$
  $\mathbf{A}_{\text{minus}} \leftarrow [\mathbf{A}_{\text{disc}}; \text{clip}(\mathbf{A}_{\text{cont}} - \boldsymbol{\Delta}, 0, 1)]$
  $\mathcal{A}_{\text{cand}} \leftarrow \mathcal{A}_{\text{cand}} \cup \{\mathbf{A}_{\text{plus}}, \mathbf{A}_{\text{minus}}\}$
end for
return $\mathcal{A}_{\text{cand}}$ ▷ Returns tensor of shape $[2 \cdot N_{\text{dir}} + 1, B, 6]$

B.3 Model Architecture

B.3.1 Embedding Layers

Definition B.4 (Spectral Normalized Embeddings)**.** *To strictly enforce Lipschitz continuity for stability guarantees (as proven in Appendix C), all linear mappings in the embedding layers utilize Spectral Normalization (SN). The weight matrix W is normalized by its largest singular value $\sigma(W)$:*

$$\overline{W}_{SN} = \frac{W}{\sigma(W)}, \quad \sigma(W) = \max_{\mathbf{x}\neq 0} \frac{\|W\mathbf{x}\|_2}{\|\mathbf{x}\|_2}. \quad \text{(B.5)}$$

For $i \in \{1, 2, \ldots, w\}$, let $\mathbf{S}_{t-w+i} = \mathbf{S}_{hist}[i, :]$ denote the i-th historical state $\big($the same applies to $\mathbf{A}_{t-w+i}\big)$. The embeddings are computed as follows, noting the explicit usage of tanh *for the*

action projection:

$$\mathbf{h}_{hist}^{(i)} = MLP_{SN}\left([\mathbf{S}_{t-w+i}; \mathbf{A}_{t-w+i}]\right) \in \mathbb{R}^d; \quad \text{(B.6)}$$

$$\mathbf{h}_t = MLP_{SN}\left(\mathbf{S}_t\right) \in \mathbb{R}^d; \quad \text{(B.7)}$$

$$\mathbf{a}_{emb} = \tanh\left(Linear_{SN}\left(\mathbf{A}_t\right)\right) \in \mathbb{R}^d, \quad \text{(B.8)}$$

where d is the model dimension (default 64; see Table 3) and MLP_{SN} denotes a Multi-Layer Perceptron with Spectral-Normalized weights.

Definition B.5 (Positional Encoding)**.** *Sinusoidal positional encodings are added to inject temporal ordering:*

$$PE\left(t, 2i\right) = \sin\left(t/10000^{2i/d}\right); \quad \text{(B.9)}$$

$$PE\left(t, 2i+1\right) = \cos\left(t/10000^{2i/d}\right), \quad \text{(B.10)}$$

for position $t \in \{1, \ldots, L_{\max}\}$ and dimension index i. The maximum sequence length is set to $L_{\max} = w + 2$ to accommodate the w historical embeddings, the current state embedding, and to provide a safety margin for the positional encoding buffer.

Definition B.6 (Sequence Formation and Context Encoding)**.** *The embedded sequence is assembled and processed as follows:*

1. ***Sequence Assembly:*** *Concatenate the history embeddings $\left\{\mathbf{h}_{hist}^{(i)}\right\}_{i=1}^{w}$ with the current state embedding $\mathbf{h}_t$:*

$$\mathbf{H}_{seq} = \left[\mathbf{h}_{hist}^{(1)}, \ldots, \mathbf{h}_{hist}^{(w)}, \mathbf{h}_t\right] \in \mathbb{R}^{(w+1)\times d}. \quad \text{(B.11)}$$

2. ***Positional Encoding Application:*** *Add sinusoidal positional encodings (Definition B.5) to inject temporal ordering:*

$$\mathbf{H}_{pos} = \mathbf{H}_{seq} + PE\left(1 : w+1\right), \quad \text{(B.12)}$$

   which allows $\mathcal{T}$ to distinguish between identical states occurring at different points in the trajectory without increasing the embedding dimensionality.

3. ***Transformer Encoding:*** *Process through the Transformer Encoder (Definitions B.7, B.8) to capture temporal dependencies (see details in Definition B.7):*

$$\mathbf{Z} = TransformerEncoder\left(\mathbf{H}_{pos}\right) \in \mathbb{R}^{(w+1)\times d}. \quad \text{(B.13)}$$

4. ***Context Extraction:*** *Extract the final hidden state as the context-aware representation:*

$$\mathbf{z}_t = \mathbf{Z}\left[w+1, :\right] \in \mathbb{R}^d. \quad \text{(B.14)}$$

   Here, $\mathbf{z}_t$ serves as the compressed temporal context, encoding the dynamics of the entire history window through self-attention.

B.3.2 Transformer Encoder

Definition B.7 (Encoder Layer Mechanics)**.** *Let $\mathbf{z}_{in}$ be the input to a layer. The layer output $\mathbf{z}_{out}$ is computed via two sub-layers (Attention and FFN) with residual connections in $TransformerEncoder\left(\cdot\right)$ of Definition B.6:*

$$\mathbf{z}' = LayerNorm\left(\mathbf{z}_{in} + MultiHead\left(\mathbf{z}_{in}, \mathbf{z}_{in}, \mathbf{z}_{in}\right)\right);$$
$$\mathbf{z}_{out} = LayerNorm\left(\mathbf{z}' + FFN\left(\mathbf{z}'\right)\right), \quad \text{(B.15)}$$

where $FFN\left(\mathbf{x}\right) = W_2 \cdot SiLU\left(W_1\mathbf{x} + b_1\right) + b_2$. Note that $\mathbf{z}_t$ in Definition B.6 corresponds to the output $\mathbf{z}_{out}$ of the N-th layer at the last sequence position.

Definition B.8 (Multi-Head Attention)**.** *With h attention heads:*

$$MultiHead\left(Q, K, V\right) = Concat\left(head_1, \ldots, head_h\right) W^O, \quad \text{(B.16)}$$

where $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$ and

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \quad \text{(B.17)}$$

B.3.3 Action Fusion

Definition B.9 (Cross-Attention Fusion)**.** *When* `use_multi_heads=True`*, action-aware fusion uses cross-attention:*

$$\mathbf{h}_{final} = LayerNorm\left[\mathbf{z}_t + MultiHead\left(\mathbf{z}_t, \mathbf{a}_{emb}, \mathbf{a}_{emb}\right)\right]. \quad \text{(B.18)}$$

Here, $\mathbf{z}_t$ and $\mathbf{a}_{emb}$ are treated as single-token sequences ($l_Q = l_K = 1$, i.e., $\mathbf{z}_t \equiv \mathbf{z}_t^{1\times d}$ and $\mathbf{a}_{emb} \equiv \mathbf{a}_{emb}^{1\times d}$), enabling cross-attention between the context representation and the action embedding.

Otherwise, using simple concatenation:

$$\mathbf{h}_{final} = \left[\mathbf{z}_t; \mathbf{a}_{emb}\right]. \quad \text{(B.19)}$$

B.3.4 Output Heads

The state vector $\mathbf{S}_t = [\mathcal{E}_t; \Phi_t; \mathbf{H}_t] \in \mathbb{R}^{11}$ comprises the error state, pressure state, and context state, as defined in Definitions A.1, A.6, and A.7.

Definition B.10 (Prediction Heads with Inertia Assumption)**.**
The dynamics model outputs predictions specifically for the Error ($\mathcal{E}$) and Context ($\mathbf{H}$) components. The Pressure (Φ) component is assumed to follow an Inertia Principle ($\Phi_{t+1} = \Phi_t$), reflecting the exogenous nature of resource constraints as discussed in Section 3.2, and is therefore not predicted by the network.

1. Component Prediction: *The latent features $\mathbf{h}_{final}$ are projected to obtain the residuals (or next states):*

$$\left[\hat{\mathcal{E}}_{out}, \hat{\mathbf{H}}_{out}\right] = Head_{shared}\left(\mathbf{h}_{final}\right). \tag{B.20}$$

If `use_separate_heads_eh=True`*, separate spectral-normalized MLPs are used for $\hat{\mathcal{E}}$ and $\hat{\mathbf{H}}$:*

$$\begin{aligned} \hat{\mathcal{E}}_{out} &= MLP_{SN}^{(\mathcal{E})}\left(\mathbf{h}_{final}\right) \in \mathbb{R}^4; \\ \hat{\mathbf{H}}_{out} &= MLP_{SN}^{(\mathbf{H})}\left(\mathbf{h}_{final}\right) \in \mathbb{R}^4. \end{aligned} \tag{B.21}$$

This separation enables the model to learn specialized representations for error and context components, improving prediction accuracy when their dynamics differ significantly.

2. Next State Assembly: *Let $\mathbf{S}_t = [\mathcal{E}_t; \Phi_t; \mathbf{H}_t]$. The predicted next state $\hat{\mathbf{S}}_{t+1}$ is constructed as:*

$$\begin{aligned} \hat{\mathcal{E}}_{t+1} &= \mathcal{E}_t + \hat{\mathcal{E}}_{out} \quad (\textit{if predict\_delta=True}); \\ \hat{\Phi}_{t+1} &= \Phi_t \qquad\qquad (\textbf{\textit{Inertia Assumption}}); \\ \hat{\mathbf{H}}_{t+1} &= \mathbf{H}_t + \hat{\mathbf{H}}_{out} \quad (\textit{if predict\_delta=True}); \\ \hat{\mathbf{S}}_{t+1} &= Concat\left(\hat{\mathcal{E}}_{t+1}, \hat{\Phi}_{t+1}, \hat{\mathbf{H}}_{t+1}\right). \end{aligned}$$

3. Cox Risk Head: *An auxiliary head predicts the log-risk of system failure based on the pre-fusion context representation $\mathbf{z}_t$ (i.e., the Transformer output before action fusion in Definition B.6). This design reflects the assumption that a trajectory's intrinsic risk is determined primarily by its historical state sequence rather than the specific action currently under consideration:*

$$\log h_t = Head_{risk}\left(\mathbf{z}_t\right) \in \mathbb{R}^1 \tag{B.22}$$

where the risk score h_t will be used in the Cox Risk Loss (Definition B.14) for survival analysis.

B.4 Training Details

B.4.1 Loss Functions

Definition B.11 (Prediction Loss)**.**
We employ component-wise Huber loss with homoscedastic uncertainty weighting and sample reweighting w_i to penalize trajectories with increasing V (Definition D.4) or high pressure.

Let B denote the batch size and $\mathcal{D}_{pred}$ the set of indices corresponding to the Error ($\mathcal{E}$) and Context ($\mathbf{H}$) components (excluding Pressure Φ, which follows the inertia assumption). The total prediction loss is:

$$\mathcal{L}_{pred} = \sum_{d \in \mathcal{D}_{pred}} \left[\exp\left(-s_d\right) \cdot \left(\frac{1}{B} \times \sum_{i=1}^{B} w_i \cdot \mathcal{L}_{Huber}\left(\hat{y}_{i,d}, y_{i,d}\right) \right) + s_d \right], \tag{B.23}$$

where s_d is the learnable log-variance parameter for dimension d, and $w_i \geqslant 1$ is the teacher reweighting factor. $\mathcal{L}_{Huber}$ is defined with $\delta = 1.0$ (consistent with `PyTorch` *default) as below:*

$$\mathcal{L}_{Huber}\left(y, \hat{y}\right) = \begin{cases} \frac{1}{2}\left(y - \hat{y}\right)^2 & |y - \hat{y}| \leqslant \delta; \\ \delta\left(|y - \hat{y}| - \frac{\delta}{2}\right) & \textit{otherwise}. \end{cases} \tag{B.24}$$

Definition B.12 (Single-Step CLF Loss)**.**
Let $V\left(\mathcal{E}\right) = \mathcal{E}^\top \mathbf{P} \mathcal{E}$ be the quadratic Lyapunov function (Definition D.4). Given state $\mathbf{S}_t$, histories $(\mathbf{S}_{hist}, \mathbf{A}_{hist})$, and candidates $\mathcal{A}_{cand} = \{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$ from Algorithm 2, the loss is:

$$\mathcal{L}_{CLF}\begin{pmatrix} \mathbf{S}_t, \mathbf{S}_{hist}, \\ \mathbf{A}_t, \mathbf{A}_{hist}, \\ \theta \end{pmatrix} = \mathbb{E}\left[ReLU\left(\sum_{k=1}^{K} w_k \Delta V_k \right) \right], \tag{B.25}$$

where $\Delta V_k = V\left(\hat{\mathcal{E}}_{t+1}^{(k)}\right) - (1-\rho)\, V\left(\mathcal{E}_t\right)$, with $V\left(\mathcal{E}\right) = \mathcal{E}^\top \mathbf{P} \mathcal{E}$ as defined in Definition D.4, and the next error state $\hat{\mathcal{E}}_{t+1}^{(k)}$ is extracted from the predicted state by the dynamics model $\mathcal{T}_\theta$ using the full history context with Φ inertia:

$$\hat{\mathbf{S}}_{t+1}^{(k)} = \mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{S}_{hist}, \mathbf{A}_k, \mathbf{A}_{hist}\right); \tag{B.26}$$

$$\hat{\mathcal{E}}_{t+1}^{(k)} = \left[\hat{\mathbf{S}}_{t+1}^{(k)}\right]_{1:4}, \tag{B.27}$$

and w_k are the softmin weights over ΔV_k:

$$w_k = \frac{\exp\left(-\Delta V_k / \tau\right)}{\sum_j \exp\left(-\Delta V_j / \tau\right)}, \tag{B.28}$$

where $\tau > 0$ is the softmin temperature (default $\tau = 0.5$), controlling the sharpness of the weighting distribution: smaller τ yields sharper focus on the best candidate, while larger one enables smoother gradient flow.

Definition B.13 (Multi-Step CLF Loss)**.** *Extending the CLF constraint to a prediction horizon H, we perform a simulated rollout. At each step j, the optimal local action is selected. The loss accumulates discounted stability violations, optionally utilizing Conditional Value at Risk (CVaR) (Rockafellar and Uryasev, 2000) to focus on tail risks (see Proposition D.5 for the theoretical justification):*

$$\mathcal{L}_{N\text{-}CLF} = \sum_{j=0}^{H-1} \gamma^j \cdot \mathcal{R}_\alpha \left(ReLU \left(\Delta V^{(j)} \right) \right), \tag{B.29}$$

where $\gamma \in (0, 1]$ is the discount factor, $\Delta V^{(j)} = \max \left(0, V \left(\hat{\mathcal{E}}_{j+1} \right) - (1 - \rho) V \left(\mathcal{E}_j \right) \right)$ represents the Lyapunov stability violation, and $\mathcal{R}_\alpha$ acts as an empirical CVaR operator: let $\mathbf{v}^+ = \{ v_i : v_i > 0 \}$ be the set of violations in a batch, and $k = \max \left(1, \lceil (1 - \alpha) \cdot |\mathbf{v}^+| \rceil \right)$; the aggregated risk is:

$$\mathcal{R}_\alpha (\mathbf{v}) = \frac{1}{k} \sum_{i=1}^{k} v_{(i)}, \tag{B.30}$$

where $v_{(i)}$ is the i-th largest element in $\mathbf{v}^+$ (we define $\mathcal{R}_\alpha (\mathbf{v}) = 0$ when $\mathbf{v}^+ = \emptyset$, i.e., all samples satisfy the CLF condition). When CVaR is disabled ($\alpha = 0$), this reduces to the empirical mean of all positive violations. This formulation directly aligns the robust training objective with the ultimate uniform boundedness (UUB) guarantees in Theorem 3.2.

Truncated Backpropagation Through Time (BPTT). *To prevent gradient explosion during the H-step rollout, we apply truncated BPTT with a window size W_{BPTT}. At rollout step j, gradients are detached for predicted states older than W_{BPTT} steps:*

$$\hat{\mathbf{S}}_{eff}^{(j-k)} = \begin{cases} \hat{\mathbf{S}}^{(j-k)} & k < W_{BPTT}; \\ detach \left(\hat{\mathbf{S}}^{(j-k)} \right) & k \geqslant W_{BPTT}. \end{cases} \tag{B.31}$$

This limits the computational graph depth while maintaining trajectory coherence within the truncation window.

Composite Multi-Step Loss. *In practice, the training objective at each rollout step combines multiple stability terms as defined in Definition B.15:*

$$\mathcal{L}_{step}^{(i)} = \mathcal{L}_{CLF}^{(i)} + \lambda_{CBF} \cdot \mathcal{L}_{CBF}^{(i)} + \lambda_{ADT} \cdot \mathcal{L}_{ADT}^{(i)}. \tag{B.32}$$

Policy Entropy Regularization. *We maximize* marginal policy entropy *over the planning horizon and batch to encourage action diversity. Let $\mathcal{H}$ be defined on the empirical distribution of selected actions:*

$$\mathcal{H}(\pi) = -\sum_{k=1}^{K} \bar{p}_k \log \left(\bar{p}_k + \varepsilon \right), \tag{B.33}$$

where $\bar{p}_k = \frac{1}{H \cdot B} \sum_{t=1}^{H} \sum_{b=1}^{B} \mathbb{1} \{ \mathbf{A}_{t,b} = k \}$ and $\varepsilon = 10^{-9}$. Here, $\bar{p}_k$ represents the global frequency of choosing action k, preventing the policy from collapsing into a single mode (e.g., always selecting "None").

The total multi-step loss includes the policy entropy regularization term $\mathcal{H}(\pi)$ to encourage action diversity:

$$\mathcal{L}_{N\text{-}step} = \sum_{i=0}^{H-1} \gamma^i \mathcal{L}_{step}^{(i)} - \lambda_{ent} \cdot \mathcal{H}(\pi). \tag{B.34}$$

This formulation ensures that the model learns not only to satisfy the Lyapunov decreasing condition (CLF), but also to maintain safety constraints (CBF) and control smoothness (ADT) throughout the simulated trajectory.

Definition B.14 (Cox Risk Loss)**.** *Using the risk scores h_t from the Cox Risk Head (Definition B.10), we define the partial likelihood loss for trajectories with failure events:*

$$\mathcal{L}_{cox} = -\frac{1}{N_{event}} \sum_{i:E_i=1} \left[\log h_i - \log \sum_{j:T_j \geqslant T_i} h_j \right], \tag{B.35}$$

where $E_i = 1$ indicates a failure event at T_i (failure event: $\|\mathcal{E}\|_2 > \theta_{event}$ with a default threshold $\theta_{event} = 2.0$).

Definition B.15 (Additional Stability Losses)**.** *These auxiliary losses complement the CLF objective to ensure safe and smooth control:*

Control Barrier Function (CBF): *Penalizes trajectories approaching resource limits. By the*

compactness of $\mathcal{S}_\Phi$ (Proposition A.3), we define $h(\mathbf{S}) = \Phi^2_{critical} - \|\Phi\|_2^2 \geqslant 0$ and the following:

$$\mathcal{L}_{CBF} = \mathbb{E}\left[ReLU\left(\left(1-\alpha_{CBF}\right) h\left(\mathbf{S}_t\right) - h\left(\mathbf{S}_{t+1}\right)\right)\right]. \tag{B.36}$$

This ensures the system remains within safe operating bounds, complementing the stability guarantee of CLF. See theoretical justifications in Proposition D.8.

Action Dissimilarity Term (ADT): *Penalizes frequent switching between candidate actions to prevent control chattering and ensure smooth policy behavior:*

$$\mathcal{L}_{ADT} = \mathbb{E}\left[\mathbb{1}\left\{k_t \neq k_{t-1}\right\}\right], \tag{B.37}$$

where $k_t \in \{1, \ldots, K\}$ is the index of the optimal action candidate selected at step t. This reduces oscillatory behavior while maintaining responsiveness to state changes.

Definition B.16 (Total Training Loss)**.** *The complete training objective combines prediction, stability, and risk estimation:*

$$\mathcal{L}_{total} = \mathcal{L}_{pred} + w_{stab} \cdot \mathcal{L}_{stab} + \lambda_{cox} \cdot \mathcal{L}_{cox}, \tag{B.38}$$

where w_{stab} follows a curriculum schedule (Definition B.18). The stability component is:

$$\mathcal{L}_{stab} = \lambda_{N\text{-}step} \cdot \mathcal{L}_{N\text{-}step} + \lambda_{jac} \cdot \mathcal{L}_{jac}. \tag{B.39}$$

When single-step mode is used instead of multi-step, $\mathcal{L}_{N\text{-}step}$ reduces to the weighted single-step CLF loss $\lambda_{CLF} \cdot \mathcal{L}_{CLF}$ (Definition B.12).

Remark B.1 (Relationship to Theoretical Stability)**.** *The theoretical stability, guaranteed by the UUB bound in Theorem D.4 and the exponential decay in Proposition D.4, depends* ***solely*** *on the CLF condition $V(\mathcal{E}_{t+1}) \leqslant (1-\rho) V(\mathcal{E}_t)$ being satisfied. The UUB proof analyzes the evolution of the Lyapunov function V, not the loss function $\mathcal{L}$. Specifically:*

- ***CLF Loss*** *($\mathcal{L}_{CLF}$): Directly enforces the Lyapunov decreasing condition. When $\mathcal{L}_{CLF} = 0$, there exists an action satisfying $\Delta V \leqslant 0$ (Proposition D.3), enabling the UUB bound derivation.*
- ***CBF, ADT, Entropy, Cox, Jacobian****: These are auxiliary training objectives that improve practical performance (safety, smoothness, exploration, risk estimation, generalization) but do* not *appear in the Lyapunov stability proofs. The theoretical guarantees hold regardless of whether these terms are enabled.*

B.4.2 Regularization

Definition B.17 (Jacobian Regularization)**.** *Penalizing the local Lipschitz constant (cf. Theorem C.3):*

$$\mathcal{L}_{jac} = \mathbb{E}\left[\sigma_{\max}\left(\frac{\partial \mathcal{T}_\theta}{\partial \mathbf{S}_t}\right)\right], \tag{B.40}$$

calculated via Singular Value Decomposition (SVD) on the exact Jacobian matrix computed by automatic differentiation.

This enforces Lipschitz continuity (Theorem C.2), enabling stability guarantees (Appendix D) by bounding $\sigma_{\max}(\nabla_{\mathbf{S}} \mathcal{T}_\theta)$.

B.4.3 Curriculum Learning Schedule

Definition B.18 (Two-Phase Training)**.** *We employ 2-phase curriculum training for stability:*

- ***Phase 1*** *$\left(\text{epochs } e = 1 \to E_1\right)$: While keeping $w_{stab}(e) = 0$, the dynamics model $\mathcal{T}$ depends on pure prediction without any stability control during this phase;*
- ***Phase 2*** *$\left(\text{epochs } e > E_1\right)$: Linear warm-up of stability weight, starting from $w_{stab}(e) = 0$ at the first epoch of Phase 2:*

$$w_{stab}(e) = \min\left[1, \frac{e - (E_1 + 1)}{E_{warmup}}\right]. \tag{B.41}$$

This ensures a smooth transition where the stability penalty is introduced gradually after the prediction manifold is established in the dynamics model $\mathcal{T}$.

Definition B.19 (Hyperparameter Annealing)**.** *Let $K = |\mathcal{A}_{cand}|$ (Algorithm 2) and $e' = \max(0, e - E_1)$ be the Phase 2 epoch count. The following components allow diverse and detailed hyperparameters control for the action selection mechanism in multi-step CLF training (Definition B.13):*

- ***CLF contraction rate ρ:*** *This component allows $\mathcal{T}$ to dynamically control the stability constraint aggressiveness in the Lyapunov violation ΔV_k (Definition B.12). It linearly interpolates from ρ_{start} (default 0.2) to ρ_{end} $\left(\text{default same as } \rho_{start}\right)$ over $E_{\rho\text{-}warmup}$ epochs*

(default 10*):*

$$\rho\left(e'\right) = \rho_{start} + \left(\rho_{end} - \rho_{start}\right) \times \min\left(1, \frac{e'}{E_{\rho\text{-}warmup}}\right). \quad \text{(B.42)}$$

Larger ρ enforces faster convergence but shrinks the feasible action space (see Section D.6);

- ***Gumbel-Softmax temperature τ_{Gumbel}:*** *Controls action selection sharpness at each rollout step.*

  When using the Gumbel-Softmax selector (`nstep_selector='gumbel_st'`)*, eq. (B.28) is replaced by a differentiable discrete sampler defined as:*

$$w_k^{(j)} = GumbelSoftmax\big(-\Delta V_1^{(j)}, \ldots, -\Delta V_K^{(j)}; \tau_{Gumbel}\big), \quad \text{(B.43)}$$

  where:

$$\Delta V_k^{(j)} = V\left(\hat{\mathcal{E}}_{t+j+1}^{(k)}\right) - (1-\rho)\, V\left(\hat{\mathcal{E}}_{t+j}^{sim}\right)$$

  is the Lyapunov violation at rollout step j for candidate $k \in \mathcal{A}_{cand}$, following the single-step formulation in Definition B.12. Exponentially decays from τ_{start} (default 1.0*) to τ_{end} (default* 0.1*) over $E_{\tau\text{-}anneal}$ epochs (default* 15*):*

$$\tau_{Gumbel}\left(e'\right) = \tau_{start} \cdot \left(\frac{\tau_{end}}{\tau_{start}}\right)^{\mathcal{Z}(e')} \quad \text{(B.44)}$$

  where $\mathcal{Z}\left(e'\right) = \min\left(1, e'/E_{\tau\text{-}anneal}\right)$. This transitions from exploration (smooth soft-selection at high τ) to exploitation (sharp near-hard selection at low τ);

- ***ϵ-greedy exploration:*** *After selecting $k_{opt}^{(j)} = \arg\min_k \Delta V_k^{(j)}$ at each rollout step j, the final action is randomized with a prob. ϵ:*

$$k_{final}^{(j)} = \begin{cases} k_{opt}^{(j)} & \text{with prob. } 1-\epsilon; \\ k_{cand}^{(j)} & \text{with prob. } \epsilon, \end{cases} \quad \text{(B.45)}$$

  where $k_{cand}^{(j)} = Uniform\left(\{1, \ldots, K\}\right)$ and ϵ linearly decay from ϵ_{start} (default 0.3*) to ϵ_{end} (default* 0.01*) over $E_{\epsilon\text{-}decay}$ epochs:*

$$\epsilon\left(e'\right) = \epsilon_{start} - \left(\epsilon_{start} - \epsilon_{end}\right) \times \min\left(1, \frac{e'}{E_{\epsilon\text{-}decay}}\right). \quad \text{(B.46)}$$

  This encourages diverse action exploration early in training to avoid local minima.

B.5 Hyperparameter Summary

Table 3: Brief training configuration of dynamics $\mathcal{T}$ (Default from `t_train_Transformer.py`)

Parameter	Symbol	Default
Model Architecture		
Model dimension	d	64
Number of layers	N	3
Attention heads	h	4
FFN hidden dimension	d_{ffn}	256
History window	w	4
Dropout rate	-	0.1
Training Schedule		
Phase 1 (predict only)	E_1	10 eps
Stability warmup	E_{warmup}	12 eps
Stability Constraints		
Initial contraction rate	ρ_{start}	0.2
Final contraction rate	ρ_{end}	ρ_{start}
Jacobian regularization	λ_{jac}	0.0
CLF softmin temperature	τ_{CLF}	0.5

Note: The history window w denotes the number of historical timesteps (excluding the current state). The total sequence length processed by the Transformer is $w+1$.

Key Design Choices.

- **History window** $w = 4$: Provides w historical timesteps for capturing temporal dependencies in decoding dynamics (Definition B.2); the Transformer processes sequences of length $w+1$, and positional encoding (Definition B.5) supports up to $L_{\max} = w + 2 = 6$ positions.
- **Curriculum learning** (Definition B.18): Pure prediction training in Phase 1 allows $\mathcal{T}$ to learn accurate dynamics before gradually introducing stability constraints (Definition B.12), avoiding conflicting gradients.
- **Stability Contraction** ($\rho = 0.2$): By default, a constant contraction rate is used to enforce Lyapunov stability (Definition D.5), though annealing is supported via Definition B.19.

B.6 Student Policy Distillation

To address the latency of online planning, we distill the teacher policy into a lightweight network π_ϕ.

Definition B.20 (Student Network Architecture)**.** *The student policy π_ϕ mirrors the encoder structure of the dynamics model $\mathcal{T}_\theta$ (Definition B.1)*

with identical hyperparameters (e.g., dimension d, layers N, heads h, history length w) as specified in Table 3. The architecture consists of:

1. ***Shared Encoder:*** *State and history embeddings (Definition B.4), positional encoding (Definition B.5), and a Transformer encoder (Definition B.7) producing context $\mathbf{z}_t \in \mathbb{R}^d$.*
2. ***Classification Head:*** *Projects $\mathbf{z}_t$ to index logits $\hat{\mathbf{l}} \in \mathbb{R}^4$ for $\{none, exact, hnsw, ivf\_pq\}$.*
3. ***Regression Head:*** *Projects $\mathbf{z}_t$ to normalized parameters $\hat{\mathbf{p}} = \left[\hat{k}_{norm}, \hat{\lambda}\right] \in [0,1]^2$ via sigmoid activation.*

The student uses the same scaler as $\mathcal{T}_\theta$ for state normalization, ensuring consistent input preprocessing.

Definition B.21 (Composite Masked Distillation Loss)**.**
*The student π_ϕ approximates the teacher's optimal action $\mathbf{A}^*$, derived from Lyapunov-guided optimization over $\mathcal{T}_\theta$ (Definition B.1). We employ a **masked learning strategy** that excludes the "none" class, preventing policy collapse and enforcing* conditional optimality*: the student masters retrieval configurations (k, λ) within the active space, maximizing $\mathcal{T}_\theta$ utilization when the safety valve permits.*

$$\mathcal{L}_{policy} = \mathcal{L}_{CE}\left(\hat{\mathbf{l}}_{[1:4]}, y^*_{idx}\right) + \beta_{reg} \cdot \mathcal{L}_{MSE}\left(\hat{\mathbf{p}}, \mathbf{p}^*\right), \tag{B.47}$$

where:

- $\mathcal{L}_{CE}$*: Cross-entropy computed over the active logits corresponding to indices $\{1,2,3\}$ (Exact, HNSW, IVF-PQ).*
- $\hat{\mathbf{l}}_{[1:4]} \in \mathbb{R}^3$*: Sliced logits corresponding to $\{exact, hnsw, ivf\_pq\}$.*
- $y^*_{idx} = \arg\max\left(\mathbf{y}^*_{[1:4]}\right) \in \{0,1,2\}$*: Re-indexed target label derived from the sliced one-hot vector, where $0 \mapsto$ exact, $1 \mapsto$ hnsw, $2 \mapsto$ ivf_pq.*
- β_{reg} *(default 1.0): Balances the classification of retrieval methods and the regression of continuous parameters.*
- $\mathbf{p}^* = [k^*/k_{\max}, \lambda^*]$*: Normalized continuous targets from the optimal action (k^*, λ^*) of the dynamics teacher model $\mathcal{T}$.*

By masking the "none" action (via `./scripts/05_train_policy_network.py`*), the student specializes in* how *to retrieve optimally when the Safety Valve permits.*

C Lipschitz Continuity: Theorems and Proofs

This appendix establishes Lipschitz continuity for the dynamics model $\mathcal{T}_\theta$ (Appendix B), providing the theoretical foundation for stability analysis.

C.1 Preliminaries

Definition C.1 (Lipschitz Continuity)**.** *A mapping $f : X \to Y$ between metric spaces (X, d_X) and (Y, d_Y) is* Lipschitz continuous *with constant $L \geqslant 0$ if, for all $x_1, x_2 \in X$, we have:*

$$d_Y\left(f\left(x_1\right), f\left(x_2\right)\right) \leqslant L \cdot d_X\left(x_1, x_2\right).$$

The optimal constant is denoted $Lip\left(f\right)$.

Remark C.1 (Physical Significance)**.** *For our dynamics model, Lipschitz continuity ensures:*

(a) bounded sensitivity to input perturbations;
(b) controllable error accumulation in iterative prediction;
(c) prerequisites for Lyapunov stability analysis (detailed in Appendix D).

C.2 Basic Component Lemmas

Lemma C.1 (Linear Mapping)**.** *For $f\left(x\right) = Wx + b$ with $W \in \mathbb{R}^{m \times n}$:*

$$Lip\left(f\right) = \|W\|_2 = \sigma_{\max}\left(W\right)$$

where $\sigma_{\max}\left(W\right)$ is the spectral norm (maximum singular value).

Proof. By the definition of matrix norms: $\|f\left(x_1\right) - f\left(x_2\right)\|_2 = \|W\left(x_1 - x_2\right)\|_2 \leqslant \|W\|_2 \|x_1 - x_2\|_2$. The bound is tight when $(x_1 - x_2)$ is the right singular vector of W corresponding to $\sigma_{\max}$. □

Lemma C.2 (SiLU Activation)**.** *The SiLU (Sigmoid-weighted Linear Unit) activation function (Elfwing et al., 2018), defined as $SiLU\left(x\right) = x \cdot \sigma\left(x\right)$ where $\sigma\left(x\right) = 1/\left(1 + e^{-x}\right)$ is the sigmoid function, satisfies $Lip\left(SiLU\right) \leqslant 1.1$.*

Proof. Computing the derivative: $\text{SiLU}'\left(x\right) = \sigma\left(x\right) + x \cdot \sigma\left(x\right)\left(1 - \sigma\left(x\right)\right)$. Numerical analysis shows $\max_x \left|\text{SiLU}'\left(x\right)\right| \approx 1.0998 \leqslant 1.1$, achieved at $x \approx 2.4$. By the Mean Value Theorem, for any $x_1, x_2 \in \mathbb{R}$, we have $\left|\text{SiLU}\left(x_1\right) - \text{SiLU}\left(x_2\right)\right| \leqslant 1.1 \times \left|x_1 - x_2\right|$. □

Lemma C.3 (ReLU Activation)**.** *The Rectified Linear Unit, defined as* $\mathit{ReLU}(x) = \max(0, x)$ *for* $x \in \mathbb{R}$*, is globally* 1*-Lipschitz continuous:* $\mathit{Lip}(\mathit{ReLU}) = 1$.

Proof. The derivative satisfies $\mathrm{ReLU}'(x) = \mathbb{1}[x > 0] \in \{0, 1\}$ almost everywhere. By the Mean Value Theorem, for any $x_1, x_2 \in \mathbb{R}$:

$$|\mathrm{ReLU}(x_1) - \mathrm{ReLU}(x_2)| \leqslant 1 \cdot |x_1 - x_2|$$

The bound is tight when both $x_1, x_2 > 0$. □

Lemma C.4 (Layer Normalization)**.** *Let* $\mathbf{x} \in \mathbb{R}^d$ *be the input vector, with a sample mean* $\mu(\mathbf{x}) = \frac{1}{d}\sum_{i=1}^{d} x_i$ *and a sample variance* $\sigma^2(\mathbf{x}) = \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2$*. Layer Normalization (Ba et al., 2016) is defined as:*

$$\mathit{LN}(\mathbf{x})_i = \gamma_i \cdot \frac{x_i - \mu(\mathbf{x})}{\sqrt{\sigma^2(\mathbf{x}) + \epsilon}} + \beta_i,$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$ *are learnable scale and shift parameters, and* $\epsilon > 0$ *is a small constant for numerical stability. On any compact set* $K \subset \mathbb{R}^d$*, where* $\sigma^2(\mathbf{x}) \geqslant \delta > 0$ *for all* $\mathbf{x} \in K$*, the mapping LN is Lipschitz continuous on* K *with*

$$\mathit{Lip}(\mathit{LN}|_K) \leqslant L_{\mathit{LN}}(K, \delta, \boldsymbol{\gamma}, \epsilon). \tag{C.1}$$

Proof. Note that the shift parameter $\boldsymbol{\beta}$ introduces a constant translation and does not affect the derivative of LN and thus the Lipschitz constant. We focus on the scaled normalization term.

Define the normalized coordinates as $\hat{x}_i = (x_i - \mu)/\sqrt{\sigma^2 + \epsilon}$. The Jacobian matrix J satisfies $J_{ij} = \partial \mathrm{LN}(\mathbf{x})_i / \partial x_j = \gamma_i \cdot \partial \hat{x}_i / \partial x_j$, since $\boldsymbol{\beta}$ is constant.

To compute $\partial \hat{x}_i / \partial x_j$, recall that $\partial \mu / \partial x_j = 1/d$ and $\partial \sigma^2 / \partial x_j = 2(x_j - \mu)/d$. Let $s = \sqrt{\sigma^2 + \epsilon}$ for brevity. Then,

$$\begin{aligned}
\frac{\partial \hat{x}_i}{\partial x_j} &= \frac{1}{s}\left(\delta_{ij} - \frac{\partial \mu}{\partial x_j}\right) - \frac{x_i - \mu}{s^2} \cdot \frac{1}{2s} \cdot \frac{\partial \sigma^2}{\partial x_j} \\
&= \frac{1}{s}\left(\delta_{ij} - \frac{1}{d}\right) - \frac{x_i - \mu}{s^2} \cdot \frac{1}{2s} \cdot \frac{2(x_j - \mu)}{d} \\
&= \frac{1}{s}\left(\delta_{ij} - \frac{1}{d}\right) - \frac{(x_i - \mu)(x_j - \mu)}{d s^3} \\
&= \frac{1}{s}\left(\delta_{ij} - \frac{1}{d} - \frac{\hat{x}_i \hat{x}_j}{d}\right),
\end{aligned} \tag{C.2}$$

where δ_{ij} is the Kronecker delta. Thus, by eq. (C.2):

$$J_{ij} = \frac{\gamma_i}{s}\left(\delta_{ij} - \frac{1}{d} - \frac{\hat{x}_i \hat{x}_j}{d}\right).$$

which, in matrix form, is equivalent to:

$$J = \frac{1}{s}\,\mathrm{diag}(\boldsymbol{\gamma})\left(I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top - \frac{1}{d}\hat{\mathbf{x}}\hat{\mathbf{x}}^\top\right),$$

where $\mathbf{1} \in \mathbb{R}^d$ is the all-ones vector.

To bound $\|J\|_2$, apply the submultiplicativity of the spectral norm and the triangle inequality:

$$\begin{aligned}
\|J\|_2 &\leqslant \frac{\|\mathrm{diag}(\boldsymbol{\gamma})\|_2}{s}\left\|I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top - \frac{1}{d}\hat{\mathbf{x}}\hat{\mathbf{x}}^\top\right\|_2 \\
&\leqslant \frac{\|\boldsymbol{\gamma}\|_\infty}{s}\left(\|I\|_2 + \frac{1}{d}\left\|\mathbf{1}\mathbf{1}^\top\right\|_2 + \frac{1}{d}\left\|\hat{\mathbf{x}}\hat{\mathbf{x}}^\top\right\|_2\right).
\end{aligned}$$

Here, $\|I\|_2 = 1$. For $\left\|\mathbf{1}\mathbf{1}^\top\right\|_2$, note that $\mathbf{1}\mathbf{1}^\top$ has rank one with eigenvalue d (corresponding to eigenvector $\mathbf{1}$) and all other eigenvalues zero; thus, its spectral norm is d. For $\left\|\hat{\mathbf{x}}\hat{\mathbf{x}}^\top\right\|_2 = \|\hat{\mathbf{x}}\|_2^2$, we compute:

$$\begin{aligned}
\|\hat{\mathbf{x}}\|_2^2 &= \sum_{i=1}^{d} \hat{x}_i^2 = \sum_{i=1}^{d} \frac{(x_i - \mu)^2}{s^2} \\
&= \frac{d\sigma^2}{s^2} = \frac{d\sigma^2}{\sigma^2 + \epsilon} \leqslant d,
\end{aligned}$$

since $\sigma^2/(\sigma^2 + \epsilon) \leqslant 1$ for $\epsilon > 0$. Therefore,

$$\begin{aligned}
\|J\|_2 &\leqslant \frac{\|\boldsymbol{\gamma}\|_\infty}{s}\left(1 + \frac{d}{d} + \frac{\|\hat{\mathbf{x}}\|_2^2}{d}\right) \\
&\leqslant \frac{\|\boldsymbol{\gamma}\|_\infty}{s}(1 + 1 + 1) \\
&= \frac{3\|\boldsymbol{\gamma}\|_\infty}{s} = \frac{3\|\boldsymbol{\gamma}\|_\infty}{\sqrt{\sigma^2 + \epsilon}}.
\end{aligned}$$

Since K is compact and $\sigma^2(\mathbf{x}) \geqslant \delta > 0$ for all $\mathbf{x} \in K$, we have $s = \sqrt{\sigma^2 + \epsilon} \geqslant \sqrt{\delta + \epsilon} > \epsilon$, ensuring a uniform lower bound on the denominator. Therefore, the Lipschitz constant on K is:

$$L_{\mathrm{LN}} = \frac{3\|\boldsymbol{\gamma}\|_\infty}{\sqrt{\delta + \epsilon}}. \tag{C.3}$$

This excludes degenerate cases (e.g., constant vectors with $\sigma^2 = 0$) where the Lipschitz constant becomes arbitrarily large. □

Lemma C.5 (Composite Mappings)**.** *For Lipschitz functions* $f : X \to Y$ *and* $g : Y \to Z$*:*

$$\mathit{Lip}(g \circ f) \leqslant \mathit{Lip}(g) \cdot \mathit{Lip}(f).$$

Proof. For any $x_1, x_2 \in X$:

$$\begin{aligned} & d_Z\left(g\left(f\left(x_1\right)\right), g\left(f\left(x_2\right)\right)\right) \\ \leqslant & \operatorname{Lip}(g) \cdot d_Y\left(f\left(x_1\right), f\left(x_2\right)\right) \\ \leqslant & \operatorname{Lip}(g) \cdot \operatorname{Lip}(f) \cdot d_X\left(x_1, x_2\right). \end{aligned}$$

□

Corollary C.1 (Multilayer Networks)**.** *By Lemma C.5, for an L-layer network $f = f_L \circ \cdots \circ f_1$, we have:*

$$\mathit{Lip}(f) \leqslant \prod_{i=1}^{L} \mathit{Lip}(f_i).$$

C.3 Spectral Normalization

Definition C.2 (Spectrally Normalized Layer)**.** *Let $W \in \mathbb{R}^{m \times n}$ be the weight matrix of a linear layer and $b \in \mathbb{R}^m$ be the bias vector. A spectrally normalized linear layer replaces W with $W_{SN} = W/\sigma_{\max}(W)$, where $\sigma_{\max}(W)$ denotes the spectral norm (maximum singular value) of W. The mapping is then defined as*

$$f_{SN}(x) = W_{SN}x + b, \tag{C.4}$$

for input $x \in \mathbb{R}^n$.

Proposition C.1 (Spectral Normalization Guarantee)**.** *Assume $\sigma_{\max}(W) > 0$. Then, $\mathit{Lip}(f_{SN}) = 1$.*

Proof. By Lemma C.1, $\operatorname{Lip}(f_{\mathrm{SN}}) = \sigma_{\max}(W_{\mathrm{SN}})$. Since $W_{\mathrm{SN}} = W/\sigma_{\max}(W)$, we have $\sigma_{\max}(W_{\mathrm{SN}}) = \sigma_{\max}(W)/\sigma_{\max}(W) = 1$. □

Corollary C.2 (Spectrally Normalized MLP)**.** *Consider an MLP composed of L spectrally normalized linear layers interspersed with activation functions having Lipschitz constants L_{σ_i} for $i = 1, \ldots, L$. Then:*

$$\mathit{Lip}(\mathit{MLP}) \leqslant \prod_{i=1}^{L} L_{\sigma_i}. \tag{C.5}$$

Proof. Each spectrally normalized linear layer has $\mathrm{Lip} = 1$ by Proposition C.1. By Corollary C.1 applied iteratively to the composition of linear layers and activations, the overall Lipschitz constant is bounded by the product of the activation constants. □

Remark C.2. *In our implementation using SiLU activations (Lemma C.2), the NodeMLP (a 2-layer network with SiLU between linear layers) satisfies $\mathit{Lip}(\mathit{NodeMLP}) \leqslant 1 \times 1.1 \times 1 = 1.1$.*

C.4 Attention Mechanism Analysis

Lemma C.6 (Spectral Norm Bound for Non-negative Row-stochastic Matrices)**.** *Let $A \in \mathbb{R}^{l \times l}$ be a non-negative row-stochastic matrix, i.e., $a_{ij} \geqslant 0$ for all i, j and $\sum_{j=1}^{l} a_{ij} = 1$ for each $i = 1, \ldots, l$. Then $\|A\|_2 \leqslant \sqrt{l}$.*

Proof. By the standard inequality for matrix norms, for any matrix A, we have

$$\|A\|_2 \leqslant \|A\|_F,$$

where $\|A\|_2$ denotes the spectral norm (the largest singular value of A) and $\|A\|_F = \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{l} a_{ij}^2}$ is the Frobenius norm.

Since A is non-negative and row-stochastic, each entry satisfies $0 \leqslant a_{ij} \leqslant 1$. For any $x \in [0, 1]$, the inequality $x^2 \leqslant x$ holds, as $x^2 - x = x(x-1) \leqslant 0$. Thus, for each row i,

$$\sum_{j=1}^{l} a_{ij}^2 \leqslant \sum_{j=1}^{l} a_{ij} = 1.$$

Summing over all rows yields

$$\|A\|_F^2 = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{ij}^2 \leqslant \sum_{i=1}^{l} 1 = l,$$

therefore $\|A\|_F \leqslant \sqrt{l}$.

Combining the inequalities, we obtain $\|A\|_2 \leqslant \|A\|_F \leqslant \sqrt{l}$. □

Lemma C.7 (Scaled Dot-Product Attention)**.** *For attention $\mathit{Attn}(Q, K, V) = \mathit{softmax}\left(QK^\top/\sqrt{d_k}\right)V$ on a compact domain with spectrally normalized projection matrices:*

$$\mathit{Lip}(\mathit{Attn}) \leqslant L_{\mathit{attn}} = \max\left(\frac{\sqrt{2}C_V C}{2\sqrt{d_k}}, \sqrt{l}\right), \tag{C.6}$$

where C_Q, C_K bound the spectral norms of Q, K, C_V bounds the Frobenius norm of V (all finite by spectral normalization and compactness), $C = \max(C_Q, C_K)$, and l is the sequence length.

Proof. We decompose attention into:

(1). score computation $S = QK^\top/\sqrt{d_k}$,
(2). softmax normalization $A = \mathrm{softmax}(S)$, and
(3). value aggregation $Y = AV$.

Using $Q_1K_1^\top - Q_2K_2^\top = (Q_1 - Q_2)\,K_1^\top + Q_2\,(K_1 - K_2)^\top$ and the submultiplicativity of the Frobenius norm ($\|AB\|_F \leqslant \|A\|_F \|B\|_2$):

$$\begin{aligned}
&\|S_1 - S_2\|_F \\
\leqslant &\frac{1}{\sqrt{d_k}}\big[\,\|Q_1 - Q_2\|_F \|K_1\|_2 + \\
&\qquad \|Q_2\|_2 \|K_1 - K_2\|_F\,\big] \\
\leqslant &\frac{C}{\sqrt{d_k}}\left(\|Q_1 - Q_2\|_F + \|K_1 - K_2\|_F\right),
\end{aligned}$$

where the second inequality follows from the bounded spectral norms $\|K_1\|_2 \leqslant C_K \leqslant C$ and $\|Q_2\|_2 \leqslant C_Q \leqslant C$. Applying the inequality $a + b \leqslant \sqrt{2}\sqrt{a^2 + b^2}$ for $a = \|Q_1 - Q_2\|_F$ and $b = \|K_1 - K_2\|_F$ (a consequence of the Cauchy-Schwarz inequality in $\mathbb{R}^2$), we have:

$$\|S_1 - S_2\|_F \leqslant \frac{\sqrt{2}C}{\sqrt{d_k}} \|[Q_1; K_1] - [Q_2; K_2]\|_F .$$

Since the softmax function ($1/2$-Lipschitz with respect to all ℓ_p norms by (Nair, 2025)) is applied row-wise and the Frobenius norm is compatible with independent row operations, we have:

$$\begin{aligned}
\|A_1 - A_2\|_F &\leqslant \frac{1}{2}\|S_1 - S_2\|_F \\
&\leqslant \frac{\sqrt{2}C}{2\sqrt{d_k}} \|[Q_1; K_1] - [Q_2; K_2]\|_F .
\end{aligned}$$

By Lemma C.6, given that the attention weights are row-stochastic with non-negative entries, their spectral norm satisfies $\|A\|_2 \leqslant \sqrt{l}$, where l is the sequence length. For the difference:

$$\begin{aligned}
&\|Y_1 - Y_2\|_F \\
=&\|A_1\,(V_1 - V_2) + (A_1 - A_2)\,V_2\|_F \\
\leqslant&\|A_1\|_2 \|V_1 - V_2\|_F + \|A_1 - A_2\|_F \|V_2\|_2 \\
\leqslant&\sqrt{l} \cdot \|V_1 - V_2\|_F + C_V \|A_1 - A_2\|_F \qquad \text{(C.7)} \\
\leqslant&\frac{\sqrt{2}C_V C}{2\sqrt{d_k}} \|[Q_1; K_1] - [Q_2; K_2]\|_F + \\
&\sqrt{l} \cdot \|V_1 - V_2\|_F , \qquad \text{(C.8)}
\end{aligned}$$

where the first inequality follows from the triangle inequality and the submultiplicativity of the Frobenius norm. Note that in eq. (C.7), we have $\|V_2\|_2 \leqslant C_V$ since $\|V\|_2 \leqslant \|V\|_F$.

The Lipschitz constant of the attention function $\text{Attn} : (Q, K, V) \to Y$ is defined as

$$\text{Lip}\,(\text{Attn}) = \sup_{\substack{Q_1 \neq Q_2,\\ K_1 \neq K_2,\\ V_1 \neq V_2}} \|Y_1 - Y_2\|_F / d,$$

with the product space metric:

$$d = \sqrt{\|[Q_1; K_1] - [Q_2; K_2]\|_F^2 + \|V_1 - V_2\|_F^2}.$$

The bound implies:

$$\begin{aligned}
&L_{\text{attn}} = \text{Lip}\,(\text{Attn}) \\
\leqslant&\alpha\,\|[Q_1; K_1] - [Q_2; K_2]\|_F / d + \\
&\beta\,\|V_1 - V_2\|_F / d, \qquad \text{(C.9)}
\end{aligned}$$

where $\alpha = \sqrt{2}C_V C / \left(2\sqrt{d_k}\right)$ and $\beta = \sqrt{l}$.

Considering the worst-case scenarios (e.g., perturbation only in $[Q; K]$ or only in V), this yields the conservative upper bound $L_{\text{attn}} = \max(\alpha, \beta)$. □

The following analysis applies to the encoder architecture defined in Definition B.7.

Proposition C.2 (Transformer Encoder Layer)**.** *Each encoder layer with spectral normalization satisfies:*

$$\textit{Lip}\,(\textit{EncoderLayer}) \leqslant L_{\textit{LN}}^2\,(1 + L_{\textit{attn}})\,(1 + L_{\textit{ffn}}).$$

Proof. For the first sublayer: $z_1 = \text{LN}\,(x + \text{MHA}\,(x))$, the residual connection gives $\text{Lip}\,(x + \text{MHA}\,(x)) \leqslant 1 + L_{\text{attn}}$. After LayerNorm: $\text{Lip}\,(z_1 | x) \leqslant L_{\text{LN}}\,(1 + L_{\text{attn}})$. Similarly, for the second sublayer with FFN. By Lemma C.5:

$$\begin{aligned}
&\text{Lip}\,(\text{EncoderLayer}) \\
\leqslant&\left[L_{\text{LN}}\,(1 + L_{\text{ffn}})\right] \cdot \left[L_{\text{LN}}\,(1 + L_{\text{attn}})\right].
\end{aligned}$$

□

Proposition C.3 (Multi-Layer Encoder)**.** *For N encoder layers:*

$$\textit{Lip}\,(\textit{Encoder}) \leqslant \left[L_{\textit{LN}}^2\,(1 + L_{\textit{attn}})\,(1 + L_{\textit{ffn}})\right]^N . \qquad \text{(C.10)}$$

Lemma C.8 (Sinusoidal Positional Encoding)**.** *The positional encoding operation $\text{PE}\,(x) = x + p$, where p is a fixed position vector, satisfies $\textit{Lip}\,(\textit{PE}) = 1$.*

Proof. For any inputs x_1, x_2:

$$\begin{aligned}
\|\text{PE}\,(x_1) - \text{PE}\,(x_2)\|_2 &= \|(x_1 + p) - (x_2 + p)\|_2 \\
&= \|x_1 - x_2\|_2 .
\end{aligned}$$

Therefore, $\text{Lip}\,(\text{PE}) = 1$. □

C.5 Standardization Transformation

Assumption C.1 (Theoretical Smoothing Assumption of Quantile Normalization)**.**
Let n be the number of linear segments in Ψ_{quant}, $\mathcal{Q}_{in} = \{q_0, \ldots, q_n\}$ be the sorted empirical quantiles (anchors), and $\mathcal{Q}_{out} = \{o_0, \ldots, o_n\}$ be the corresponding fixed standard normal quantiles where $o_i = \Psi^{-1}\left(\frac{i+\epsilon}{n+2\epsilon}\right)$.

In raw empirical data, strict ties such as $(q_{i+1} = q_i)$ can occur for discrete features $\left(e.g., \epsilon_t^{(cov)}\right.$ defined in Definition A.3$\left.\right)$, theoretically yielding infinite slopes.

However, for the purpose of stability analysis, we model the underlying physical dynamics as continuous variables subject to intrinsic measurement noise.

*We therefore analyze the **Strictly Monotonic Surrogate**, assuming minimal quantile separation $\delta_{\min} > 0$ (theoretically equivalent to infinitesimal jitter $\xi \sim \mathcal{N}(0, \nu)$, omitted in implementation for reproducibility).*

Theorem C.1 (Lipschitz Continuity of Hybrid Standardization)**.** *The hybrid standardization $\Psi : \mathbb{R}^{11} \to \mathbb{R}^{11}$ (Definition A.8) is Lipschitz continuous on the compact state space $\mathcal{X}$, satisfying $\|\Psi(\mathbf{x}) - \Psi(\mathbf{y})\|_2 \leqslant L_\Psi \|\mathbf{x} - \mathbf{y}\|_2$ for a finite constant L_Ψ.*

Proof. The hybrid standardization Ψ operates component-wise on the 11-dimensional state space, applying either affine standardization Ψ_{std} or quantile normalization Ψ_{quant} to each dimension independently. Based on the distributional analysis in Section 3.1 and the formal component definitions in Definitions A.2–A.7, the components are partitioned as:

- Ψ_{std} (Affine) applied to unimodal/Gaussian-like components: $\epsilon_t^{(\text{sem})}$, $H_t^{(\text{con})}$, $H_t^{(\text{stab})}$, $H_t^{(\text{vol})}$, $\epsilon_t^{(\text{surp})}$, $\epsilon_t^{(\text{rep})}$.
- Ψ_{quant} (Quantile) applied to multi-modal/bounded components: $\epsilon_t^{(\text{cov})}$, $\phi_t^{(\text{lat})}$, $\phi_t^{(\text{mem})}$, $\phi_t^{(\text{thr})}$, $H_t^{(\text{foc})}$.

Since Ψ is a concatenation of independent scalar transformations, its Lipschitz constant L_Ψ with respect to the Euclidean norm is determined by the maximum component-wise Lipschitz constant:

$$L_\Psi = \max_{1 \leqslant i \leqslant 11} \text{Lip}\left(\Psi_i\right). \tag{C.11}$$

We derive the bounds for both transformation types below. We assume the state space $\mathcal{X} \subset \mathbb{R}^{11}$ is compact (closed and bounded). This compactness is guaranteed because $\mathcal{X}$ is the Cartesian product of bounded subspaces defined in Propositions A.2 ($\mathcal{E}_t$), A.3 (Φ_t), and A.4 ($\mathbf{H}_t$).

1. Affine Standardization (Ψ_{std}): For components approximating a Gaussian distribution, the transformation is $\Psi_{\text{std}}(x) = (x - \mu)/\sigma$, where μ and σ are the empirical mean and standard deviation computed over the training corpus. For any $x, y \in \mathbb{R}$, the distance scales linearly:

$$\left|\Psi_{\text{std}}(x) - \Psi_{\text{std}}(y)\right| = \left|\frac{x - y}{\sigma}\right| = \frac{1}{\sigma}|x - y|.$$

Thus, $\text{Lip}(\Psi_{\text{std}}) = 1/\sigma$. Crucially, the compactness of $\mathcal{X}$ and the non-degenerate nature of the state components (e.g., semantic drift variance is non-zero, per Proposition A.1) ensure that $\sigma \geqslant \sigma_{\min} > 0$. Therefore, $L_{\text{std}} = \max_j(1/\sigma_j) < \infty$.

2. Quantile Normalization (Ψ_{quant}): Our implementation employs `QuantileTransformer` of `sklearn` (see `t_train_Transformer.py`), which maps input features to a standard normal distribution via piecewise linear interpolation.

Under Assumption C.1, Ψ_{quant} is a continuous, piecewise linear function. Its Lipschitz constant is the maximum absolute slope across all n linear segments:

$$\text{Lip}\left(\Psi_{\text{quant}}\right) = \max_{0 \leqslant i < n} \left|\frac{o_{i+1} - o_i}{q_{i+1} - q_i}\right| \leqslant \frac{\Delta o_{\max}}{\delta_{\min}}. \tag{C.12}$$

Since $\Delta o_{\max}$ is bounded (by the regularization ϵ) and $\delta_{\min} > 0$ (by the smoothing assumption), we have $L_{\text{quant}} < \infty$.

Conclusion: Combining both cases, the global Lipschitz constant $L_\Psi = \max\left(L_{\text{std}}, L_{\text{quant}}\right)$ is finite, confirming that the input standardization layer preserves the Lipschitz continuity of the overall system. □

Remark C.3 (Theoretical Worst-Case vs. Empirical Reality)**.** *It is worth noting the distinction between the theoretical upper bound and the effective behavior of the standardization layer:*

*(i) **Theoretical Bound ($L_\Psi \gg 1$):** The constant L_Ψ can be large (empirically ≈ 126), corresponding to steep slopes in Ψ_{quant} at*

high-density regions where $q_{i+1} \approx q_i$ (Assumption C.1).

(ii) ***Effective Local Contraction:*** *In practice, the system operates on the data manifold where mappings are well-behaved. The dynamics model $\mathcal{T}_\theta$ compensates for this scaling, yielding $\mathbb{E}[\sigma_{\max}(J_\mathcal{T})] \approx 1.02$ (Proposition C.4).*

While the finiteness of L_Ψ (Theorem C.1) is sufficient to satisfy the global existence conditions for Lyapunov stability (Theorem 3.2), the local stability governing the system's runtime behavior is dictated by the much tighter empirical bounds observed in the active state regions.

C.6 Main Theorem

Theorem C.2 (Lipschitz Continuity of Dynamics Model $\mathcal{T}_\theta$)**.** *The complete dynamics model $\mathcal{T}_\theta$, configured with Spectral Normalization (`use_spectral_norm=True`), mapping the full input $\mathbf{X}$ to the next state components (predicting $\hat{\mathcal{E}}_{t+1}, \hat{\mathbf{H}}_{t+1}$ while preserving Φ_t via inertia), is globally Lipschitz continuous on the **original** compact input space (incorporating the standardization Ψ):*

$$\left\|\mathcal{T}_\theta\left(\mathbf{X}_1\right) - \mathcal{T}_\theta\left(\mathbf{X}_2\right)\right\|_2 \leqslant L_\mathcal{T} \left\|\mathbf{X}_1 - \mathbf{X}_2\right\|_2, \tag{C.13}$$

where the Lipschitz constant $L_\mathcal{T}$ of the overall dynamics model $\mathcal{T}$ is defined as:

$$L_\mathcal{T} = L_\Psi \cdot \sqrt{L_\mathcal{N}^2 + 1^2} < \infty, \tag{C.14}$$

with $L_\mathcal{N}$ defined in eq. (C.22) and the Smoothing Assumption of $L\left(\Psi_{quant}\right)$ in L_Ψ (Assumption C.1).

Proof. We decompose the dynamics model $\mathcal{T}_\theta$ into five sequential phases and bound the Lipschitz constant step-by-step, utilizing the composition property of Lipschitz mappings (Lemma C.5):

Phase 1: Historical Sequence Embedding Following the input representation in Definition B.2, the history input is formed by concatenating the state sequence $\mathbf{S}_{\text{hist}}$ and the action sequence $\mathbf{A}_{\text{hist}}$.

For non-trivial history windows ($w > 1$), and conservatively assuming the input metric is the maximum of component norms (yielding a looser bound than the Euclidean isometry), the concatenation operation has $\text{Lip}\left(\text{cat}\right) \leqslant \sqrt{2}$. This vector is processed by a 2-layer MLP (`history_seq_embed`). Since all linear layers use spectral normalization ($\text{Lip} \leqslant 1$, Proposition C.1) and the SiLU activation has $\text{Lip} \leqslant 1.1$ (Lemma C.2), the MLP satisfies $\text{Lip}\left(\text{MLP}\right) \leqslant 1 \times 1.1 \times 1 = 1.1$ (Corollary C.2).

Thus:

$$\text{Lip}\left(\text{hist\_embed}\right) \leqslant \sqrt{2} \times 1.1 \approx 1.56. \tag{C.15}$$

Phase 2: Current State Embedding The current state $\mathbf{S}_t$ is encoded by an independent 2-layer spectrally normalized MLP (`state_embed_t`):

$$\text{Lip}\left(\text{state\_embed}\right) \leqslant \text{Lip}\left(\text{MLP}\right) \leqslant 1.1. \tag{C.16}$$

Phase 3: Transformer Encoding The input sequence is formed by concatenating historical and current embeddings ($\text{Lip} = 1$ for concatenation along the time dimension) and adding sinusoidal positional encodings ($\text{Lip} = 1$, Lemma C.8). The sequence is processed by an N-layer Transformer encoder. By Proposition C.3, and noting that the Euclidean norm of the flattened state vector is isometric to the Frobenius norm of the sequence matrix (validating the direct application of L_{attn} from Lemma C.7), the Lipschitz constant L_{enc} for one layer is bounded by $L_{\text{LN}}^2\left(1 + L_{\text{attn}}\right)\left(1 + L_{\text{ffn}}\right)$. For N layers:

$$\begin{aligned}\text{Lip}\left(\text{Transformer}\right) &\leqslant \left(L_{\text{enc}}\right)^N \\ &= \left[L_{\text{LN}}^2\left(1 + L_{\text{attn}}\right)\left(1 + L_{\text{ffn}}\right)\right]^N.\end{aligned} \tag{C.17}$$

We extract the final hidden state via a linear projection ($\text{Lip} \leqslant 1$).

Phase 4: Action and Auxiliary Fusion Following Definition B.9, we fuse the Transformer output $\mathbf{z}_t$ with the action $\mathbf{A}_t$ (Definition B.3) and optional auxiliary inputs $\mathbf{I}_{\text{aux}}$.

- **Action Embedding:** $\mathbf{a}_{\text{emb}} = \tanh\left(W\mathbf{A}_t\right)$. Since the derivative $|\tanh'| \leqslant 1$ and the weight matrix is spectrally normalized ($\|W\|_2 \leqslant 1$), the mapping is non-expansive: $\text{Lip}\left(\text{a\_emb}\right) \leqslant 1$.
- **Fusion Mechanism:** We employ Multi-Head Attention (MHA) with h parallel heads to fuse $\mathbf{z}_t$ (Query) and $\mathbf{a}_{\text{emb}}$ (Key/Value). The derivation proceeds in four strict steps:
  (i) **Projection:** Input projection matrices W^Q, W^K, W^V are spectrally normalized ($\text{Lip} \leqslant 1$). Noting that for the single-step fusion vectors $\mathbf{z}_t$ and $\mathbf{a}_{\text{emb}}$, the Frobenius

norm coincides with the Euclidean norm ($\|\cdot\|_F = \|\cdot\|_2$), the single-head sensitivity is directly governed by Lemma C.7: $\text{Lip}(\text{head}_i) \leqslant 1 \times 1 \times L_{\text{attn}} = L_{\text{attn}}$.

(ii) **Concatenation:** The outputs of h heads are concatenated. In the Euclidean (L_2) space, $\|\text{Concat}(\mathbf{y}_1, \ldots, \mathbf{y}_h)\|_2 = \sqrt{\sum \|\mathbf{y}_i\|_2^2}$. This implies a Lipschitz expansion of exactly $\sqrt{h}$ over the single-head bound.

(iii) **Aggregation:** The final output projection W^O is spectrally normalized (Lemma C.1), contributing a factor of Lip $\leqslant 1$. Combining (i)-(iii): $\text{Lip}(\text{MHA}) \leqslant 1 \cdot \sqrt{h} \cdot L_{\text{attn}}$.

(iv) **Residual & Norm:** The residual connection adds the identity mapping (Lip $= 1$), bounded by the triangle inequality as $1+\text{Lip}(\text{MHA})$. Finally, LayerNorm applies a multiplicative scaling L_{LN} (Lemma C.5).

Combining these steps yields the rigorous bound:

$$\text{Lip}(\text{fusion}) \leqslant L_{\text{LN}} \left(1 + \sqrt{h} L_{\text{attn}}\right). \quad \text{(C.18)}$$

- **Auxiliary Inputs:** Optional inputs $\mathbf{I}_{\text{aux}}$ (e.g., source embeddings, decoder hidden states) are processed by independent feature extractors. Each extractor is explicitly implemented as a 2-layer spectrally normalized MLP. By Corollary C.2 (Spectrally Normalized MLP), for each auxiliary branch j:

$$\text{Lip}(\text{aux}_j) \leqslant 1 \times \text{Lip}(\text{SiLU}) \times 1 \approx 1.1. \quad \text{(C.19)}$$

- **Feature Concatenation:** The model concatenates m distinct feature branches (the fusion output plus auxiliary embeddings). For L_2-norm inputs, the Lipschitz constant of the concatenation is bounded by the Euclidean norm of the individual branch constants: $\text{Lip}(\text{cat}) \leqslant \sqrt{\sum (\text{Lip}(\text{branch}_i))^2}$. Since the concatenation operates on orthogonal subspaces in the ℓ_2 metric, the effective Lipschitz constant is determined by the Euclidean norm of the component constants $\sqrt{\sum_i \text{Lip}(\text{branch}_i)^2}$. Comparing the contributions of the two branch types:
  - Fusion Branch:

$$L_{\text{fusion}} \approx L_{\text{LN}} \left(1 + \sqrt{h} \cdot L_{\text{attn}}\right) > 2; \quad \text{(C.20)}$$

  - Auxiliary Branch: $L_{\text{aux}} \approx 1.1$.

The effective constant for Phase 4 is:

$$\text{Lip}(\text{Phase 4}) = \sqrt{L_{\text{fusion}}^2 + \sum_j L_{\text{aux},j}^2}$$
$$\approx \sqrt{\left[L_{\text{LN}}\left(1 + \sqrt{h} L_{\text{attn}}\right)\right]^2 + m_{\text{aux}}(1.1)^2}. \quad \text{(C.21)}$$

Phase 5: Output Prediction The combined features pass through the output MLP head (Separated or Shared, $L = 2$), satisfying $\text{Lip}(\text{OutputHead}) \leqslant 1.1$.

Overall Constant: Let $L_{\mathcal{N}}$ denote the Lipschitz constant of the neural prediction path, derived from the chain composition of Phases 1 through 5:

$$L_{\mathcal{N}} = 1.56 \cdot 1.1 \cdot L_{\text{enc}}^N \cdot \text{Lip}(\text{Phase 4}) \cdot 1.1. \quad \text{(C.22)}$$

The full next state $\hat{\mathbf{S}}_{t+1}$ is constructed by concatenating the neural predictions (for $\mathcal{E}, \mathbf{H}$) and the inertia-based pressure state ($\Phi_{t+1} = \Phi_t$, implying an identity map with Lip $= 1$). Under the ℓ_2 norm, the global Lipschitz constant $L_{\mathcal{T}}$ combines the pre-processing scaling L_Ψ and the parallel output structure:

$$L_{\mathcal{T}} = L_\Psi \cdot \sqrt{L_{\mathcal{N}}^2 + 1^2} < \infty. \quad \text{(C.23)}$$

This structural decomposition confirms that the inclusion of the inertia assumption does not compromise the finiteness of the system's Lipschitz constant. □

Corollary C.3 (Residual Dynamics)**.** *When the model is configured with residual learning (`predict_delta=True`), as specified in Definition B.10, the mapping becomes* $\mathbf{S}_{t+1} = \mathbf{S}_t + \mathcal{T}_\theta^\Delta(\mathbf{S}_t, \mathbf{A}_t)$. *Since the identity map has* $Lip(Id) = 1$ *and* $\mathcal{T}_\theta^\Delta$ *has a Lipschitz constant* $L_{\mathcal{T}^\Delta}$*, the triangle inequality yields:*

$$Lip(\mathbf{S}_t \mapsto \mathbf{S}_{t+1}) \leqslant Lip(Id) + Lip(\mathcal{T}_\theta^\Delta)$$
$$= 1 + L_{\mathcal{T}^\Delta}. \quad \text{(C.24)}$$

This bound preserves the finite stability properties required for robust control, ensuring that the residual formulation does not introduce unbounded expansion.

Remark C.4 (Role in Stability Robustness)**.** *This Lipschitz constant* $L_{\mathcal{T}}$ *is formally utilized in Proposition D.6 to derive the robustness bounds of the Control Lyapunov Function against action perturbations.*

C.7 Jacobian Regularization

Theorem C.3 (Jacobian-Lipschitz Relationship)**.** *For a C^1 smooth mapping $f : \mathcal{C} \to \mathbb{R}^m$ defined on a convex domain $\mathcal{C} \subseteq \mathbb{R}^n$:*

$$Lip\left(f|_{\mathcal{C}}\right) = \sup_{\mathbf{x} \in \mathcal{C}} \sigma_{\max}\left(J_f\left(\mathbf{x}\right)\right), \tag{C.25}$$

where $J_f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}}$ is the Jacobian matrix and $\sigma_{\max}(\cdot)$ denotes the spectral norm.

Proof. **Upper bound ($\leqslant$):** Consider two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$. Since $\mathcal{C}$ is convex, the line segment $\mathbf{x}(t) = \mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)$ for $t \in [0, 1]$ lies entirely within $\mathcal{C}$. By the Fundamental Theorem of Calculus for vector-valued functions:

$$f(\mathbf{x}_1) - f(\mathbf{x}_2) = \int_0^1 J_f(\mathbf{x}(t))(\mathbf{x}_1 - \mathbf{x}_2)\, dt.$$

Applying the integral norm inequality and the definition of the induced matrix 2-norm $\|A\|_2 = \sigma_{\max}(A)$:

$$\begin{aligned} &\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_2 \\ \leqslant &\int_0^1 \|J_f(\mathbf{x}(t))(\mathbf{x}_1 - \mathbf{x}_2)\|_2\, dt \\ \leqslant &\int_0^1 \sigma_{\max}(J_f(\mathbf{x}(t)))\, \|\mathbf{x}_1 - \mathbf{x}_2\|_2\, dt \\ \leqslant &\left[\sup_{\mathbf{z} \in \mathcal{C}} \sigma_{\max}(J_f(\mathbf{z}))\right] \|\mathbf{x}_1 - \mathbf{x}_2\|_2. \end{aligned} \tag{C.26}$$

This establishes $\text{Lip}(f) \leqslant \sup_{\mathbf{x} \in \mathcal{C}} \sigma_{\max}(J_f(\mathbf{x}))$.

Lower bound ($\geqslant$): Let L be the Lipschitz constant of f on $\mathcal{C}$. By definition, $\|f(\mathbf{y}) - f(\mathbf{x})\|_2 \leqslant L\|\mathbf{y} - \mathbf{x}\|_2$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{C}$.

Fix any $\mathbf{x}_0 \in \mathcal{C}$ and let $\mathbf{v}$ be the right singular vector of $J_f(\mathbf{x}_0)$ corresponding to the largest singular value $\sigma_{\max}$. Consider $\mathbf{y} = \mathbf{x}_0 + h\mathbf{v}$ for sufficiently small $h > 0$. By the definition of the Fréchet derivative:

$$\begin{aligned} &\lim_{h \to 0} \frac{\|f(\mathbf{x}_0 + h\mathbf{v}) - f(\mathbf{x}_0)\|_2}{\|h\mathbf{v}\|_2} \\ =&\frac{\|J_f(\mathbf{x}_0)\mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sigma_{\max}(J_f(\mathbf{x}_0)). \end{aligned} \tag{C.27}$$

Since the ratio is bounded by L for all h, taking the limit implies $L \geqslant \sigma_{\max}(J_f(\mathbf{x}_0))$. Since this holds for any $\mathbf{x}_0 \in \mathcal{C}$, we have $L \geqslant \sup_{\mathbf{x} \in \mathcal{C}} \sigma_{\max}(J_f(\mathbf{x}))$. □

Proposition C.4 (Empirical Local Lipschitz Bound)**.** *For the trained Champion model, we estimate the effective Lipschitz constant of the **Learned Neural Dynamics** $\mathcal{T}_\theta : \mathbf{S}_t \to \left[\hat{\mathcal{E}}_{t+1}, \hat{\mathbf{H}}_{t+1}\right]$ by computing Jacobian spectral norm statistics on the validation set.*

*We compute the **Exact Spectral Norm** via SVD (`torch.linalg.svdvals()` in `./scripts/t_train_Transformer.py`):*

- ***Mean:*** $\mathbb{E}[\sigma_{\max}(J_\mathcal{T})] = 1.0192$ *(Standard Deviation $\sigma = 0.0253$)*
- ***99th Percentile:*** $P_{99}(\sigma_{\max}) = 1.0887$. *This confirms that for 99% of the sampled state space, the local expansion of the neural component is bounded by ≈ 1.09.*
- ***Maximum Observed:*** $\max(\sigma_{\max}) = 1.2511$ *(occurring in rare outlier states).*
- ***Method:*** *Exact SVD on the partial Jacobian matrix of the neural predictions.*

Verification Protocol. The validation suite (`s8_robust_spectral_norm_jacobian()` in `t_train_Transformer.py`) samples $N_{\text{restart}} \times N_{\text{sample}}$ points from the validation set. Statistics are computed during the post-training S8 validation phase and persisted to `summary.json`.

For each sample $\mathbf{S}_t \in \mathbb{R}^{11}$, we differentiate the neural mapping function to obtain the partial Jacobian matrix $J_\mathcal{T} = \partial\left[\hat{\mathcal{E}}_{t+1}, \hat{\mathbf{H}}_{t+1}\right]/\partial \mathbf{S}_t \in \mathbb{R}^{8 \times 11}$ using `torch.autograd.functional.jacobian()`.

The exact spectral norm $\sigma_{\max} = \|J_\mathcal{T}\|_2$ is then derived via `torch.linalg.svdvals()`. By Theorem C.2, aligning with the Inertia Assumption in Definition B.10, the global Lipschitz constant $L_\mathcal{T}$ is bounded by L_Ψ and $\sqrt{L_\mathcal{N}^2 + 1}$ (where 1 arises from the orthogonal inertia mapping of Φ_t); thus, verifying $\sigma_{\max}(J_\mathcal{T}) \approx 1$ is sufficient to constrain the system stability. □

D Lyapunov Stability: Theorems and Proofs

This appendix provides a rigorous stability analysis of the PAEC closed-loop system using Lyapunov theory.

D.1 Foundations of Lyapunov Stability

D.1.1 Discrete-Time Stability

Definition D.1 (Discrete-Time Dynamical System)**.** *Consider the autonomous system:*

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t), \quad \mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n. \tag{D.1}$$

An equilibrium $\mathbf{x}^* = f(\mathbf{x}^*)$ *is:*

- ***Stable****:* $\forall \epsilon > 0, \exists \delta > 0$ *such that* $\|\mathbf{x}_0 - \mathbf{x}^*\| < \delta \Rightarrow \|\mathbf{x}_t - \mathbf{x}^*\| < \epsilon, \forall t \geqslant 0$*;*
- ***Asymptotically stable****: Stable and* $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^*$*;*
- ***Exponentially stable****:* $\|\mathbf{x}_t - \mathbf{x}^*\| \leqslant C \cdot r^t \|\mathbf{x}_0 - \mathbf{x}^*\|$ *for some* $C > 0, r \in (0,1)$.

Definition D.2 (Lyapunov Function)**.** *A function* $V : \mathcal{X} \to \mathbb{R}$ *is a* Lyapunov function *for the equilibrium* $\mathbf{x}^*$ *if:*

(i) $V(\mathbf{x}^*) = 0$ *and* $V(\mathbf{x}) > 0$ *for* $\mathbf{x} \neq \mathbf{x}^*$ *(positive definiteness);*
(ii) $V(\mathbf{x}) \to \infty$ *as* $\|\mathbf{x}\| \to \infty$ *(radial unboundedness);*
(iii) $\Delta V(\mathbf{x}) = V(f(\mathbf{x})) - V(\mathbf{x}) \leqslant 0$ *(non-increasing along trajectories).*

This follows the classical Lyapunov direct method (Lyapunov, 1992) for analyzing stability without explicitly solving system equations.

Theorem D.1 (Lyapunov Stability Theorem (Discrete))**.** *If there exists a Lyapunov function* V *for system* $\mathbf{x}_{t+1} = f(\mathbf{x}_t)$*, then the equilibrium is stable. If additionally* $\Delta V(\mathbf{x}) < 0$ *for* $\mathbf{x} \neq \mathbf{x}^*$*, it is asymptotically stable. The proof follows standard Lyapunov arguments (Khalil, 2002, Chapter 4).*

D.1.2 Control Lyapunov Functions

Definition D.3 (Control Lyapunov Function)**.** *For a controlled system* $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$*, a function* V *is a* Control Lyapunov Function (CLF) *if, for every* $\mathbf{x} \neq \mathbf{0}$*, there exists* $\mathbf{u} \in \mathcal{U}$ *such that:*

$$V(f(\mathbf{x}, \mathbf{u})) < V(\mathbf{x}). \tag{D.2}$$

The CLF concept was introduced by Artstein (1983); Sontag (1989) provided the universal formula for constructing stabilizing feedback. See Freeman and Kokotovic (1996) for the comprehensive treatment of CLF-based robust control.

Theorem D.2 (CLF and Stability)**.** *If* V *is a CLF and the policy* π *satisfies* $\forall\, \mathbf{x}_t \neq \mathbf{0}$*,*

$$V(f(\mathbf{x}_t, \pi(\mathbf{x}_t))) \leqslant (1-\rho) \cdot V(\mathbf{x}_t) \tag{D.3}$$

for some $\rho \in (0,1)$*, then the closed-loop system is exponentially stable:*

$$V(\mathbf{x}_t) \leqslant (1-\rho)^t \cdot V(\mathbf{x}_0). \tag{D.4}$$

Proof. We proceed by induction on t.

- For the base case, we have

$$V(\mathbf{x}_0) \leqslant (1-\rho)^0 V(\mathbf{x}_0) = V(\mathbf{x}_0),$$

  which holds trivially.
- For the inductive step, assume that eq. (D.4) holds. By the CLF condition, we have:

$$\begin{aligned} V(\mathbf{x}_{t+1}) &\leqslant (1-\rho) \cdot V(\mathbf{x}_t) \\ &\leqslant (1-\rho)^{t+1} \cdot V(\mathbf{x}_0). \end{aligned}$$

□

D.2 PAEC Lyapunov Framework

D.2.1 Lyapunov Candidate Function

Definition D.4 (Quadratic Lyapunov Function for PAEC)**.** *We define the Lyapunov candidate function:*

$$V(\mathcal{E}) = \mathcal{E}^\top \mathbf{P} \mathcal{E}, \tag{D.5}$$

where $\mathbf{P} = \mathit{diag}(p_{sem}, p_{cov}, p_{surp}, p_{rep}) \succ 0$ *is a positive-definite diagonal weight matrix. By default,* $p_i = 1.0$ *for all* i*. The error state* $\mathcal{E} \in \mathbb{R}^4$ *is defined in Definition A.1.*

Proposition D.1 (Properties of V)**.** $V(\mathcal{E})$ *satisfies:*

(i) ***Positive definiteness:*** $V(\mathcal{E}) \geqslant p_{\min} \|\mathcal{E}\|_2^2 > 0$ *for* $\mathcal{E} \neq \mathbf{0}$*;*
(ii) ***Quadratic bounds:*** $p_{\min} \|\mathcal{E}\|_2^2 \leqslant V(\mathcal{E}) \leqslant p_{\max} \|\mathcal{E}\|_2^2$*;*
(iii) ***Radial unboundedness:*** $V(\mathcal{E}) \to \infty$ *as* $\|\mathcal{E}\| \to \infty$*,*

where $p_{\min} = \min_i p_i$ *and* $p_{\max} = \max_i p_i$.

Proof. Since $\mathbf{P} = \operatorname{diag}(p_1, \ldots, p_4)$ with $p_i > 0$, for all $\mathcal{E} \neq \mathbf{0}$, we have:

$$V(\mathcal{E}) = \sum_i p_i \epsilon_i^2 \geqslant p_{\min} \sum_i \epsilon_i^2 = p_{\min} \|\mathcal{E}\|_2^2 > 0.$$

For the upper bound, we have:

$$V(\mathcal{E}) = \sum_i p_i \epsilon_i^2 \leqslant p_{\max} \|\mathcal{E}\|_2^2 .$$

If $\|\mathcal{E}\| \to \infty$, there exists at least one $|\epsilon_i| \to \infty$ such that $V(\mathcal{E}) \geqslant p_{\min} \epsilon_i^2 \to \infty$. □

Proposition D.2 (Properties of Learnable **P** Matrix)**.** *When* **P** *is parameterized via:*

$$P_{ii} = \textit{softplus}\left[(p_{\textit{param}})_i\right] + \epsilon, \tag{D.6}$$

with $\epsilon = 10^{-6}$*, we have:*

(i) ***Positive definiteness:*** **P** *is strictly positive definite for any* $\mathbf{p}_{\textit{param}} \in \mathbb{R}^4$*, ensuring* $V(\mathcal{E})$ *remains a valid Lyapunov candidate.*

(ii) ***Bounded gradients:*** *The gradient of the Lyapunov energy with respect to the learnable parameters,* $\partial V / \partial (p_{\textit{param}})_i$*, is bounded, preventing gradient explosion during the optimization of the dynamics model.*

Proof. (i) The softplus function is strictly positive by $\text{softplus}(x) = \log(1 + e^x) > 0$. With the stability term ϵ, the diagonal entries in eq. (D.6) satisfy that:

$$P_{ii} = \text{softplus}\left[(p_{\text{param}})_i\right] + \epsilon \geqslant \epsilon > 0, \ \forall i.$$

Since **P** is diagonal with strictly positive entries, it is positive definite.

(ii) By the chain rule:

$$\frac{\partial V}{\partial (p_{\text{param}})_i} = \left(\epsilon^{(i)}\right)^2 \cdot \frac{\partial \text{softplus}}{\partial (p_{\text{param}})_i} = \left(\epsilon^{(i)}\right)^2 \cdot \sigma\left((p_{\text{param}})_i\right),$$

where $\sigma(x) \in (0, 1)$ is the sigmoid function.

As established in Definition A.1 and Proposition A.1, the error state components are physically bounded (e.g., semantic drift $\epsilon^{(\text{sem})} \in [0, 2]$). Letting $C_{\max} = \max_i \left(\epsilon_{\max}^{(i)}\right)^2$, we have:

$$\left| \frac{\partial V}{\partial (p_{\text{param}})_i} \right| \leqslant C_{\max} \cdot 1. \tag{D.7}$$

For our specific state definition, $C_{\max} = 4$ in eq. (D.7), ensuring that the gradients remain well-behaved. □

D.2.2 Lyapunov Decreasing Condition

Definition D.5 (CLF Condition for PAEC)**.** *Given the convergence rate* $\rho \in (0, 1)$*, the* CLF decreasing condition *is defined as:*

$$V(\mathcal{E}_{t+1}) \leqslant (1 - \rho) \cdot V(\mathcal{E}_t); \tag{D.8}$$

or equivalently:

$$\Delta V_t = V(\mathcal{E}_{t+1}) - (1 - \rho) \cdot V(\mathcal{E}_t) \leqslant 0. \tag{D.9}$$

Remark D.1 (Exponential Decay)**.** *When the CLF condition holds at each step, by eq. (D.4):*

$$V(\mathcal{E}_t) \leqslant (1 - \rho)^t \cdot V(\mathcal{E}_0), \tag{D.10}$$

i.e., the error energy $V(\mathcal{E}_t)$ *decays exponentially at rate* $(1 - \rho)$*.*

D.3 CLF Loss and Optimization

Proposition D.3 (CLF Loss Optimization Objective)**.** *Minimizing* $\mathcal{L}_{\textit{CLF}}$ *in Definition B.12 encourages the existence of an action* $\mathbf{A}_k \in \mathcal{A}_{\textit{cand}}$ *such that*

$$V(\mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A}_k)_{\mathcal{E}}) \leqslant (1 - \rho) \cdot V(\mathcal{E}_t) \tag{D.11}$$

for most states in the training distribution.

Proof. The ReLU activates only when the weighted average $\sum_k w_k \Delta V_k > 0$. The softmin weighting focuses on actions with the smallest ΔV_k. With a small temperature τ, the softmin function approximates the minimum operator $(\min_k \Delta V_k)$. Thus, if there exists an action satisfying $(\min \Delta V_k \leqslant 0)$, the weighted sum will be non-positive, yielding zero loss. □

Definition D.6 (Multi-Step CLF Loss (Theoretical))**.** *For theoretical analysis, we define the multi-step CLF loss over the horizon* H *with a discount* γ*:*

$$\mathcal{L}_{\textit{N-CLF}} = \sum_{i=0}^{H-1} \gamma^i \mathcal{L}_{\textit{CLF}}^{(i)}. \tag{D.12}$$

At each step i*, the action that minimizes* ΔV *is selected greedily.*

This captures the pure CLF *component; the complete loss including CBF, ADT, and entropy is specified in Definition B.13.*

Remark D.2 (Prerequisite of Multi-Step CLF Stability Analysis)**.** *For Proposition D.4, we assume greedy action selection* $\mathbf{A}_{t+i}^{sim} = \arg\min_{\mathbf{A}} \Delta V$ *at each rollout step* i*. Auxiliary terms (CBF, ADT, entropy) from Definition B.15 are excluded but do not affect UUB stability (Proposition D.8).*

Proposition D.4 (Multi-Step CLF Stability)**.** *If the single-step CLF condition* $V(\mathcal{E}_{t+1}) \leqslant (1-\rho) V(\mathcal{E}_t)$ *holds for all* t*, then infinite-horizon asymptotic stability follows, with finite training horizon* H *sufficing for generalization. In the ideal case (accurate* $\mathcal{T}_\theta$*), if* $\mathcal{L}_{CLF}^{(i)} = 0$ *for* $i \in \{0, \ldots, H-1\}$ *and greedy action* $\mathbf{A}_{t+i}^{sim} = \arg\min_{\mathbf{A}} \Delta V$*, then:*

$$V\left(\mathcal{E}_{t+H}^{sim}\right) \leqslant (1-\rho)^H V(\mathcal{E}_t). \tag{D.13}$$

By Remark D.2, this bound depends only on single-step $\mathcal{L}_{CLF}^{(i)} = 0$*, independent of the aggregation of the other loss terms.*

Proof. By Proposition D.3, $\mathcal{L}_{\text{CLF}}^{(i)} = 0$ implies $\exists \mathbf{A}$ with $\Delta V \leqslant 0$; greedy selects it. Recursively:

$$\begin{aligned} V\left(\mathcal{E}_{t+1}^{\text{sim}}\right) &\leqslant (1-\rho) \cdot V(\mathcal{E}_t) \\ V\left(\mathcal{E}_{t+2}^{\text{sim}}\right) &\leqslant (1-\rho) \cdot V\left(\mathcal{E}_{t+1}^{\text{sim}}\right) \\ &\leqslant (1-\rho)^2 \cdot V(\mathcal{E}_t) \\ &\vdots \\ V\left(\mathcal{E}_{t+H}^{\text{sim}}\right) &\leqslant (1-\rho)^H \cdot V(\mathcal{E}_t). \end{aligned}$$

For infinite t, $V(\mathcal{E}_t) \leqslant (1-\rho)^t \cdot V(\mathcal{E}_0) \to 0$ as $0 < 1-\rho < 1$, yielding asymptotic stability; finite H verifies and generalizes via recursion. □

D.4 Closed-Loop System Stability

D.4.1 System Definition

Definition D.7 (PAEC Closed-Loop System)**.** *The complete system with implicit policy:*

$$\begin{cases} \mathbf{A}_t = \pi_\theta(\mathbf{S}_t) = \arg\min_{\mathbf{A} \in \mathcal{A}_{cand}} V\left(\mathcal{E}_{t+1}^{(\mathbf{A})}\right); \\ [\mathcal{E}_{t+1}, \mathbf{H}_{t+1}] = \mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{hist}, \mathbf{A}_{hist}); \\ \Phi_{t+1} = \Phi_t \quad \textit{(pressure inertia)}, \end{cases} \tag{D.14}$$

where $\mathbf{S}_{hist} = [\mathbf{S}_{t-w}, \ldots, \mathbf{S}_{t-1}]$ *and* $\mathbf{A}_{hist} = [\mathbf{A}_{t-w}, \ldots, \mathbf{A}_{t-1}]$ *denote the sliding-window history context with window size* w*, as specified in Definition B.1.*

D.4.2 Ideal Case: Exponential Stability

We first establish the ideal-case stability result under the following assumptions, which will be systematically relaxed in Section D.4.4.

Assumption D.1 (Model Perfection)**.** *The learned dynamics model* $\mathcal{T}_\theta$ *exactly predicts the true system dynamics, i.e.,* $\mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A}_t)_{\mathcal{E}} = \mathcal{E}_{t+1}^{true}$ *for all* $\mathbf{S}_t \in \mathcal{S}$ *and* $\mathbf{A}_t \in \mathcal{A}$*.*

Assumption D.2 (CLF Feasibility)**.** *For every state* $\mathbf{S}_t$ *in the operational domain* $\mathcal{S}$*, there exists at least one feasible action* $\mathbf{A} \in \mathcal{A}_{cand}$ *such that the CLF contraction condition (Definition D.5) is satisfied:*

$$V\left(\mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A})_{\mathcal{E}}\right) \leqslant (1-\rho) \cdot V(\mathcal{E}_t). \tag{D.15}$$

Assumption D.3 (Policy Optimality)**.** *The implicit policy* π_θ *(eq. (3.12) in the main text) successfully selects an action satisfying the CLF condition whenever such an action exists.*

Theorem D.3 (Exponential Stability – Ideal Case)**.** *Under Assumptions D.1–D.3, by Theorem D.2, the PAEC closed-loop system (Definition D.7) achieves exponential asymptotic stability:*

$$V(\mathcal{E}_t) \leqslant (1-\rho)^t \cdot V(\mathcal{E}_0) \xrightarrow{t\to\infty} 0. \tag{D.16}$$

Proof. We verify that the closed-loop system satisfies the CLF condition required by Theorem D.2. By Assumption D.2, for any state $\mathbf{S}_t$, there exists an action $\mathbf{A} \in \mathcal{A}_{\text{cand}}$ such that eq. (D.15) is satisfied.

By Assumption D.3, the policy π_θ selects such an action: $\mathbf{A}_t = \pi_\theta(\mathbf{S}_t)$. By Assumption D.1, the predicted next state equals the true state:

$$\mathcal{E}_{t+1}^{\text{true}} = \mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A}_t)_{\mathcal{E}}. \tag{D.17}$$

Combining these, the closed-loop system satisfies $V(\mathcal{E}_{t+1}) \leqslant (1-\rho) \cdot V(\mathcal{E}_t)$ for all $t \geqslant 0$. By Theorem D.2, this implies exponential decay:

$$V(\mathcal{E}_t) \leqslant (1-\rho)^t \cdot V(\mathcal{E}_0).$$

Since $\rho \in (0,1)$, we have $0 < 1-\rho < 1$, and thus $\lim_{t\to\infty} (1-\rho)^t = 0$. By the sandwich theorem with $V(\mathcal{E}_t) \geqslant 0$ (Proposition D.1), we conclude $V(\mathcal{E}_t) \to 0$ as $t \to \infty$. □

D.4.3 Probabilistic Stability under Stochastic Noise

The ideal-case result (Theorem D.3) assumes perfect model fidelity and deterministic dynamics—conditions that are inevitably violated in production environments. Real-world deployments introduce two fundamental sources of uncertainty:

(1). the learned dynamics model $\mathcal{T}_\theta$ incurs prediction errors, and
(2). external disturbances (traffic spikes, hardware variability, concurrent workloads) inject stochastic noise into the system.

To derive practically meaningful stability guarantees, we systematically relax the ideal assumptions as follows:

1. **Model Perfection → Bounded Error:** We replace Assumption D.1 (exact dynamics prediction) with Assumption D.4, which permits bounded prediction errors $\|\boldsymbol{\delta}_t\| \leqslant \delta_{\max}$. This reflects the inherent approximation error of any learned model.
2. **Deterministic → Stochastic Dynamics:** We introduce Assumption D.5 to model external disturbances $\boldsymbol{\xi}_t$ as a zero-mean stochastic process with finite covariance. This captures production-level uncertainties absent from the deterministic ideal case.
3. **CLF on True → CLF on Predicted Dynamics:** Assumptions D.2–D.3 jointly imply that the policy achieves CLF contraction on the *true* dynamics. In the presence of model error, we retain this property on *predicted* dynamics via Assumption D.6, which the CLF loss (Definition B.12) directly enforces during training.

Under these relaxed assumptions, the true PAEC dynamics decompose as follows:

$$\mathcal{E}_{t+1}^{\text{true}} = \underbrace{\mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}_t\right)_{\mathcal{E}}}_{\text{predicted (CLF satisfied)}} + \underbrace{\boldsymbol{\delta}_t}_{\text{model error}} + \underbrace{\boldsymbol{\xi}_t}_{\text{system noise}}. \tag{D.18}$$

We now formally define these three assumptions and establish the theoretical grounding that bridges engineering implementation with control-theoretic guarantees.

Assumption D.4 (Bounded Model Error)**.** *There exists a constant $\delta_{\max} > 0$ such that for all states, the deterministic model prediction error is bounded:*

$$\left\|\mathcal{E}_{t+1}^{true} - \mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}_t\right)_{\mathcal{E}}\right\|_2 \leqslant \delta_{\max}. \tag{D.19}$$

The following proposition, by the multi-step CLF loss (Definition B.13), establishes that CVaR-based training improves robustness by reducing the worst-case model error $\delta_{\max}$ in Assumption D.4, tightening the UUB bound in Theorem D.4.

Proposition D.5 (Robustness via CVaR)**.** *Minimizing the Conditional Value-at-Risk (CVaR) loss*

$$\mathcal{L}_{CVaR} = \mathbb{E}\left[\mathcal{L}_{CLF} \mid \mathcal{L}_{CLF} > VaR_\alpha\left(\mathcal{L}_{CLF}\right)\right] \tag{D.20}$$

focuses optimization on the worst $(1-\alpha)$ fraction of samples, thereby reduces the worst-case model error $\delta_{\max}$ in the tail of the distribution.

Proof. Recall the variational definition of CVaR (Rockafellar and Uryasev, 2000):

$$\text{CVaR}_\alpha\left(\mathcal{L}\right) = \min_\nu \left\{\nu + \frac{1}{1-\alpha}\mathbb{E}\left[(\mathcal{L}-\nu)^+\right]\right\}. \tag{D.21}$$

The gradient of this loss with respect to model parameters θ is:

$$\nabla_\theta \mathcal{L}_{\text{CVaR}} = \frac{1}{1-\alpha}\mathbb{E}\left[\mathop{\mathbb{1}}_{\mathcal{L}>\nu^*} \nabla_\theta \mathcal{L}\right], \tag{D.22}$$

where ν^* is the Value-at-Risk (VaR) threshold. This formulation reveals that optimization is performed *exclusively* on the subset of samples where the loss exceeds ν^* (the worst $1-\alpha$ fraction).

Since high CLF loss is strongly correlated with large prediction errors $\|\boldsymbol{\delta}_t\|$, this mechanism effectively re-weights the training distribution to focus on reducing $\delta_{\max}$ (the error in the most difficult regions). Given that the UUB stability bound scales as $b_\delta \propto \delta_{\max}^2$ (as will be established in Theorem D.4), minimizing the tail risk provides a quadratic improvement in the rigorous safety guarantee.

By explicitly penalizing the tail of the loss distribution, the optimizer is forced to improve model fidelity specifically in these high-error regions, thereby reducing $\delta_{\max}$ and quadratically tightening the UUB bound b_δ. □

Assumption D.5 (Finite-Variance System Noise)**.** *The system noise $\boldsymbol{\xi}_t$ is a stochastic process with zero mean and finite covariance $\boldsymbol{\Sigma}_\xi = \mathbb{E}\left[\boldsymbol{\xi}_t \boldsymbol{\xi}_t^\top\right]$. We do not assume a bounded support (e.g., Gaussian tails are permitted), ensuring the generality of Theorem D.4.*

Notice that the finite-variance requirement in Assumption D.5 is minimal, accommodating heavy-tailed distributions common in production systems. To validate that PAEC maintains stability under realistic conditions, we instantiate $\boldsymbol{\xi}_t$ with a challenging traffic model in the following Remark D.3.

Remark D.3 (Engineering Instantiation for Stress Testing)**.** *While Theorem D.4 holds for any finite-variance noise, our experimental validation employs a* ***high-fidelity traffic model*** *to stress-test the controller. Specifically, the* `ProductionConstraintSimulator.py` *generates $\boldsymbol{\xi}_t$ via a composite process:*

1. ***Base Load:*** *$\mu_t \sim \Gamma(\alpha, \beta)$ simulating long-tail request distributions;*
2. ***Diurnal Cycles:*** *A sinusoidal component $A \cdot \sin(2\pi t/T)$ capturing daily traffic tides;*
3. ***Jitter:*** *Additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ for high-frequency volatility.*

This complex instantiation verifies that the PAEC controller maintains stability even under non-stationary, non-Gaussian disturbances typical of real-world deployments.

The finite-variance condition enables concentration bounds. Corollary D.1 bounds $\|\boldsymbol{\xi}_t\|_2$ with probability $1-\delta$, supporting Theorem D.4.

Lemma D.1 (Multidimensional Chebyshev Inequality)**.** *Let $\mathbf{X} \in \mathbb{R}^d$ be a random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For any $\epsilon > 0$:*

$$\mathbb{P}\left\{\|\mathbf{X}-\boldsymbol{\mu}\|_2 \geqslant \epsilon\right\} \leqslant \frac{Tr\left(\boldsymbol{\Sigma}\right)}{\epsilon^2}. \tag{D.23}$$

Proof. Let $\mathbf{Y} = \mathbf{X} - \boldsymbol{\mu}$. Since $\|\mathbf{Y}\|_2 \geqslant \epsilon$ implies $\|\mathbf{Y}\|_2^2/\epsilon^2 \geqslant 1$:

$$\mathbb{1}_{\{\|\mathbf{Y}\|_2 \geqslant \epsilon\}} \leqslant \frac{\|\mathbf{Y}\|_2^2}{\epsilon^2}.$$

Taking expectations:

$$\mathbb{P}\left\{\|\mathbf{Y}\|_2 \geqslant \epsilon\right\} = \mathbb{E}\left[\mathbb{1}_{\{\|\mathbf{Y}\|_2 \geqslant \epsilon\}}\right] \leqslant \frac{1}{\epsilon^2}\mathbb{E}\left[\|\mathbf{Y}\|_2^2\right].$$

By the linearity of expectation:

$$\mathbb{E}\left[\|\mathbf{Y}\|_2^2\right] = \mathbb{E}\left[\sum_{i=1}^{d} Y_i^2\right] = \sum_{i=1}^{d}\mathbb{E}\left[Y_i^2\right]$$

$$= \sum_{i=1}^{d} \mathrm{Var}\left(X_i\right) = \mathrm{Tr}\left(\boldsymbol{\Sigma}\right).$$

Substituting yields the result. □

Corollary D.1 (Probabilistic Noise Bound)**.** *For any confidence level $\delta \in (0,1)$, defining $\xi_{\max,\delta} = \sqrt{Tr\left(\boldsymbol{\Sigma}_\xi\right)/\delta}$ ensures:*

$$\mathbb{P}\left\{\|\boldsymbol{\xi}_t\|_2 \geqslant \xi_{\max,\delta}\right\} \leqslant \delta. \tag{D.24}$$

Proof. By Assumption D.5, $\boldsymbol{\xi}_t$ has zero mean ($\boldsymbol{\mu} = \mathbf{0}$) and covariance $\boldsymbol{\Sigma}_\xi$. Applying Lemma D.1 with $\mathbf{X} = \boldsymbol{\xi}_t$ and $\boldsymbol{\mu} = \mathbf{0}$:

$$\mathbb{P}\left\{\|\boldsymbol{\xi}_t\|_2 \geqslant \epsilon\right\} \leqslant \frac{\mathrm{Tr}\left(\boldsymbol{\Sigma}_\xi\right)}{\epsilon^2}.$$

Setting the probability upper bound to δ and solving for ϵ:

$$\frac{\mathrm{Tr}\left(\boldsymbol{\Sigma}_\xi\right)}{\epsilon^2} = \delta \implies \epsilon = \sqrt{\frac{\mathrm{Tr}\left(\boldsymbol{\Sigma}_\xi\right)}{\delta}} = \xi_{\max,\delta}.$$

□

With the probabilistic bound on system noise established in Corollary D.1, we now introduce the last assumption required for the Ultimate Uniform Boundedness result. This assumption ensures that the control policy satisfies the Lyapunov decreasing condition on average.

Assumption D.6 (CLF Condition on Predicted Dynamics)**.** *The learned policy π_θ satisfies the CLF contraction condition (Definition D.5) on the predicted dynamics:*

$$V\left(\hat{\mathcal{E}}_{t+1}\right) = V\left(\mathcal{T}_\theta\left(\mathbf{S}_t, \pi_\theta\left(\mathbf{S}_t\right)\right)_{\mathcal{E}}\right)$$

$$\leqslant (1-\rho)\cdot V\left(\mathcal{E}_t\right), \tag{D.25}$$

for all states $\mathbf{S}_t$ in the operational domain $\mathcal{S}$.

Remark D.4 (Relationship to Ideal Case Assumptions)**.** *Assumption D.6 follows from Assumptions D.2–D.3 on predicted dynamics. The CLF loss (Definition B.12) enforces this condition during training.*

D.4.4 Ultimate Uniform Boundedness (UUB)

Having established the bounded model error (Assumption D.4), finite-variance noise (Assumption D.5), and the average CLF condition (Assumption D.6), we now provide the complete step-by-step derivation of the Ultimate-Uniform-Boundedness (UUB) bound for the closed-loop system under the actual dynamics.

Theorem D.4 (Probabilistic Ultimate Uniform Boundedness)**.** *Under Assumptions D.4–D.6, the closed-loop error energy satisfies:*

$$\begin{aligned}&\limsup_{t\to\infty} V\left(\mathcal{E}_t\right) \leqslant b_\delta \\ \approx &\frac{2p_{\max}\eta_{\max,\delta}^2}{\rho}\left[1+\frac{2p_{\max}\left(1-\rho\right)}{\rho p_{\min}}\right], \end{aligned} \tag{D.26}$$

with a probability of at least $1-\delta$*. Here:*

- $\eta_{\max,\delta} = \delta_{\max} + \xi_{\max,\delta}$ *is the aggregate disturbance bound;*
- $p_{\max} = \lambda_{\max}\left(\mathbf{P}\right)$ *and* $p_{\min} = \lambda_{\min}\left(\mathbf{P}\right)$ *are the maximum and minimum eigenvalues (spectral norms) of the Lyapunov matrix.*

Proof. Let $\hat{\mathcal{E}}_{t+1} = \mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}_t\right)_{\mathcal{E}}$ be the model prediction. The true state is:

$$\mathcal{E}_{t+1} = \hat{\mathcal{E}}_{t+1} + \boldsymbol{\eta}_t, \quad \text{where } \boldsymbol{\eta}_t = \boldsymbol{\delta}_t + \boldsymbol{\xi}_t.$$

By the Assumption D.4 and Assumption D.5, $\left\|\boldsymbol{\eta}_t\right\|_2 \leqslant \eta_{\max,\delta}$ with probability $1-\delta$.

Expanding the Lyapunov function, we have:

$$\begin{aligned} V\left(\mathcal{E}_{t+1}\right) =& \mathcal{E}_{t+1}^\top \mathbf{P} \mathcal{E}_{t+1} \\ =& \left(\hat{\mathcal{E}}_{t+1}+\boldsymbol{\eta}_t\right)^\top \mathbf{P}\left(\hat{\mathcal{E}}_{t+1}+\boldsymbol{\eta}_t\right) \\ =& \underbrace{\hat{\mathcal{E}}_{t+1}^\top \mathbf{P} \hat{\mathcal{E}}_{t+1}}_{\text{Term 1: Prediction}} + \underbrace{2\hat{\mathcal{E}}_{t+1}^\top \mathbf{P}\boldsymbol{\eta}_t}_{\text{Term 2: Cross-Term}} \\ &+ \underbrace{\boldsymbol{\eta}_t^\top \mathbf{P}\boldsymbol{\eta}_t}_{\text{Term 3: Noise Energy}}. \end{aligned} \tag{D.27}$$

Step 1: Analyzing Term 1 (Contraction)
By Assumption D.6, the policy π_θ achieves CLF contraction on the predicted dynamics:

$$\text{Term 1} = V\left(\hat{\mathcal{E}}_{t+1}\right) \leqslant \left(1-\rho\right) V\left(\mathcal{E}_t\right). \tag{D.28}$$

Step 2: Analyzing Disturbance Terms
Using the Cauchy-Schwarz Inequality and the definition of the operator norm $\left\|\mathbf{P}\right\|_2 = p_{\max}$ for Term 2:

$$\begin{aligned} \left|2\hat{\mathcal{E}}_{t+1}^\top \mathbf{P}\boldsymbol{\eta}_t\right| &\leqslant 2\left\|\hat{\mathcal{E}}_{t+1}\right\| \cdot p_{\max} \cdot \left\|\boldsymbol{\eta}_t\right\| \\ &\leqslant 2p_{\max}\eta_{\max,\delta}\left\|\hat{\mathcal{E}}_{t+1}\right\|. \end{aligned}$$

We apply **Young's Inequality** $\left(2ab \leqslant \epsilon a^2 + \frac{1}{\epsilon}b^2\right)$ with $a = \left\|\hat{\mathcal{E}}_{t+1}\right\|$ and $b = p_{\max}\eta_{\max,\delta}$. For any $\epsilon > 0$:

$$\text{Term 2} \leqslant \epsilon\left\|\hat{\mathcal{E}}_{t+1}\right\|^2 + \frac{p_{\max}^2\eta_{\max,\delta}^2}{\epsilon}. \tag{D.29}$$

Using the Rayleigh quotient property $p_{\min}\left\|\mathbf{x}\right\|^2 \leqslant \mathbf{x}^\top\mathbf{P}\mathbf{x} = V\left(\mathbf{x}\right)$, we have:

$$\begin{aligned} \left\|\hat{\mathcal{E}}_{t+1}\right\|^2 &\leqslant V\left(\hat{\mathcal{E}}_{t+1}\right)/p_{\min} \\ &\leqslant \left(1-\rho\right)\cdot V\left(\mathcal{E}_t\right)/p_{\min}. \end{aligned} \tag{D.30}$$

Substituting eq. (D.30) back to eq. (D.29), we get:

$$\text{Term 2} \leqslant \frac{\epsilon\left(1-\rho\right)}{p_{\min}} V\left(\mathcal{E}_t\right) + \text{Const}_{\max}, \tag{D.31}$$

where we define $\text{Const}_{\max} = p_{\max}^2 \cdot \eta_{\max,\delta}^2/\epsilon$.

To ensure net contraction, we choose ϵ such that the coefficient of $V\left(\mathcal{E}_t\right)$ absorbs a fraction of the decay rate ρ. Let the "borrowed" decay be $\rho/2$:

$$\frac{\epsilon\left(1-\rho\right)}{p_{\min}} = \frac{\rho}{2} \implies \epsilon = \frac{\rho p_{\min}}{2\left(1-\rho\right)}. \tag{D.32}$$

Substituting eq. (D.32) back into $\text{Const}_{\max}$ yields:

$$\begin{aligned} \text{Const}_{\max} =& \frac{p_{\max}^2\eta_{\max,\delta}^2}{\frac{\rho p_{\min}}{2(1-\rho)}} \\ =& \frac{2p_{\max}^2\left(1-\rho\right)}{\rho p_{\min}} \cdot \eta_{\max,\delta}^2. \end{aligned} \tag{D.33}$$

Term 3 represents the raw energy injected by noise. Take eq. (D.33) into Term 3 and use the matrix norm property:

$$\begin{aligned} \text{Term 3} &= \boldsymbol{\eta}_t^\top \mathbf{P}\boldsymbol{\eta}_t \\ &\leqslant p_{\max}\left\|\boldsymbol{\eta}_t\right\|^2 \leqslant p_{\max}\eta_{\max,\delta}^2. \end{aligned} \tag{D.34}$$

Step 3: Recursive Inequality Construction
Substituting eq. (D.28), eq. (D.31), and eq. (D.34) into eq. (D.27), we get:

$$\begin{aligned} V\left(\mathcal{E}_{t+1}\right) \leqslant& \underbrace{\left(1-\rho\right)V\left(\mathcal{E}_t\right)}_{\text{Term 1}} + \underbrace{\left(\frac{\rho}{2}V\left(\mathcal{E}_t\right) + \text{Const}_{\max}\right)}_{\text{Term 2 bound}} \\ &+ \underbrace{p_{\max}\eta_{\max,\delta}^2}_{\text{Term 3 bound}} \\ =& \left(1-\rho+\frac{\rho}{2}\right)V\left(\mathcal{E}_t\right) \\ &+ \left(\text{Const}_{\max} + p_{\max}\eta_{\max,\delta}^2\right) \\ =& \left(1-\frac{\rho}{2}\right)V\left(\mathcal{E}_t\right) + C_{\text{total}}, \end{aligned} \tag{D.35}$$

where the total disturbance constant C_{total} is:

$$\begin{aligned} C_{\text{total}} &= p_{\max}\eta_{\max,\delta}^2 + \frac{2p_{\max}^2\left(1-\rho\right)}{\rho p_{\min}}\eta_{\max,\delta}^2 \\ &= p_{\max}\eta_{\max,\delta}^2\left[1+\frac{2p_{\max}\left(1-\rho\right)}{\rho p_{\min}}\right]. \end{aligned} \tag{D.36}$$

Step 4: Ultimate Bound

We have a linear recurrence inequality of the form $x_{t+1} \leqslant (1-\tilde{\rho})\, x_t + C$, where $\tilde{\rho} = \rho/2$. The asymptotic limit is given by the sum of the geometric series:

$$\limsup_{t\to\infty} V(\mathcal{E}_t) \leqslant \frac{C_{\text{total}}}{\tilde{\rho}} = \frac{2}{\rho} \cdot C_{\text{total}}. \tag{D.37}$$

Substituting eq. (D.36) into eq. (D.37) yields the final bound:

$$b_\delta = \frac{2p_{\max}\eta_{\max,\delta}^2}{\rho}\left[1 + \frac{2p_{\max}(1-\rho)}{\rho p_{\min}}\right], \tag{D.38}$$

which completes the proof. □

D.4.5 Robustness Analysis

Proposition D.6 (CLF Robustness under Action Perturbation)**.** *Let $\mathbf{A}_t^*$ be the optimal action and $\tilde{\mathbf{A}}_t$ be a perturbed action with $\left\|\tilde{\mathbf{A}}_t - \mathbf{A}_t^*\right\|_2 \leqslant \nu_{\max}$. If the dynamics model has a Lipschitz constant $L_{\mathcal{T}}$ (Theorem C.2), then:*

$$\Delta V_{perturb} \leqslant -\rho \cdot V(\mathcal{E}_t) + L_V L_{\mathcal{T}} \nu_{\max}, \tag{D.39}$$

where $L_V = 2p_{\max}\sqrt{(1-\rho)\cdot V(\mathcal{E}_t)/p_{\min}}$.

Proof. Let $\hat{\mathcal{E}}_{t+1} = \mathcal{T}_\theta(\mathbf{S}_t, \mathbf{A}_t^*)_{\mathcal{E}}$ and $\tilde{\mathcal{E}}_{t+1} = \mathcal{T}_\theta\left(\mathbf{S}_t, \tilde{\mathbf{A}}_t\right)_{\mathcal{E}}$. By Lipschitz continuity:

$$\left\|\tilde{\mathcal{E}}_{t+1} - \hat{\mathcal{E}}_{t+1}\right\|_2 \leqslant L_{\mathcal{T}}\left\|\tilde{\mathbf{A}}_t - \mathbf{A}_t^*\right\|_2 \leqslant L_{\mathcal{T}}\nu_{\max}.$$

The change in the Lyapunov function is bounded by:

$$\begin{aligned}
&\left|V\left(\tilde{\mathcal{E}}_{t+1}\right) - V\left(\hat{\mathcal{E}}_{t+1}\right)\right| \\
\leqslant &\left\|\nabla_{\mathcal{E}} V\left(\hat{\mathcal{E}}_{t+1}\right)\right\|_2 \cdot \left\|\tilde{\mathcal{E}}_{t+1} - \hat{\mathcal{E}}_{t+1}\right\|_2 \\
= &\left\|2\mathbf{P}\hat{\mathcal{E}}_{t+1}\right\|_2 \cdot L_{\mathcal{T}}\nu_{\max} \\
\leqslant &2p_{\max}\left\|\hat{\mathcal{E}}_{t+1}\right\|_2 \cdot L_{\mathcal{T}}\nu_{\max}.
\end{aligned}$$

Using $\left\|\hat{\mathcal{E}}_{t+1}\right\|_2 \leqslant \sqrt{(1-\rho)\cdot V(\mathcal{E}_t)/p_{\min}}$ from the CLF condition completes the proof. □

D.5 Convergence Time Analysis

Having established the Ultimate Uniform Boundedness (UUB) guarantee in Theorem D.4, we now analyze the *time* at which the dynamics system converges to the bounded region.

Definition D.8 (ε-Stabilization Time)**.** *The time to reach error level ε is defined as:*

$$T_\varepsilon = \min\{t : V(\mathcal{E}_t) \leqslant \varepsilon\}. \tag{D.40}$$

Proposition D.7 (Minimum Convergence Time)**.** *In the ideal case (no noise), achieving stabilization to the error level ε requires at least*

$$T_\varepsilon \geqslant -\frac{\ln(V(\mathcal{E}_0)/\varepsilon)}{\ln(1-\rho)} \tag{D.41}$$

steps. For example, with $V_0 = 1.0$, $\varepsilon = 0.01$, and $\rho = 0.1$, we have $T_\varepsilon \geqslant 44$ steps.

Proof. From the exponential stability condition (Theorem D.3), we have $V(\mathcal{E}_t) \leqslant (1-\rho)^t V(\mathcal{E}_0)$. Setting $(1-\rho)^T V_0 \leqslant \varepsilon$, we take the natural logarithm:

$$T\ln(1-\rho) \leqslant \ln(\varepsilon/V_0) = -\ln(V_0/\varepsilon). \tag{D.42}$$

Since $0 < 1-\rho < 1$, the term $\ln(1-\rho)$ is negative. Dividing by it reverses the inequality:

$$T \geqslant \frac{-\ln(V_0/\varepsilon)}{\ln(1-\rho)} = -\frac{\ln(V_0/\varepsilon)}{\ln(1-\rho)}. \tag{D.43}$$

□

D.6 Design Trade-offs and Parameter Sensitivity

The probabilistic UUB bound derived in Theorem D.4 (cf. eq. (D.38)) reveals critical trade-offs governing controller design:

$$b_\delta \approx \frac{2p_{\max}\eta_{\max,\delta}^2}{\rho}\left[1 + \frac{2p_{\max}(1-\rho)}{\rho p_{\min}}\right]. \tag{D.44}$$

1. **Model Precision:** The bound scales with $\eta_{\max,\delta}^2$. Reducing $\delta_{\max}$ yields quadratic stability improvements.
2. **Convergence Rate (ρ):** The bound scales approximately with $O\left(1/\rho^2\right)$ for small ρ.
   - **Larger** ρ (aggressive control) rapidly reduces the theoretical error bound b_δ and convergence time T_ε.
   - **However**, large ρ shrinks feasible actions, potentially trading translation quality for stability.
3. **Curriculum Learning:** We start with small ρ to allow the model to learn dynamics with a loose stability constraint, then gradually increase ρ to tighten the bound b_δ.

D.7 Stability under Suboptimal Policies

Finally, we address the stability of the distilled "Offline Student" policy, which may not perfectly match the optimal "Online Teacher" action.

Theorem D.5 (ϵ-Suboptimality Stability)**.** *Assume the deployed policy π_ϕ yields a suboptimal action $\tilde{\mathbf{A}}_t$ with a Lyapunov gap ϵ_{sub} relative to the optimal action $\mathbf{A}^*$:*

$$V\left(\mathcal{T}_\theta\left(\mathbf{S}_t, \tilde{\mathbf{A}}_t\right)_{\mathcal{E}}\right) \leqslant V\left(\mathcal{T}_\theta\left(\mathbf{S}_t, \mathbf{A}^*\right)_{\mathcal{E}}\right) + \epsilon_{sub}. \tag{D.45}$$

If the optimal action satisfies the contraction $V(\dots) \leqslant (1-\rho)\cdot V(\mathcal{E}_t)$, then the system remains Ultimately Uniformly Bounded.

Proof. Substituting eq. (D.45) into the contraction inequality:

$$V(\mathcal{E}_{t+1}) \leqslant (1-\rho)\, V(\mathcal{E}_t) + \epsilon_{\text{sub}}.$$

This is a linear recurrence relation with a constant disturbance term ϵ_{sub}. Solving it recursively:

$$V(\mathcal{E}_t) \leqslant (1-\rho)^t V(\mathcal{E}_0) + \epsilon_{\text{sub}} \sum_{k=0}^{t-1} (1-\rho)^k. \tag{D.46}$$

Taking the limit of eq. (D.46) as $t \to \infty$, the geometric series $\sum (1-\rho)^k$ converges to $1/\rho$. Thus, eq. (D.46) is equivalent to:

$$\limsup_{t\to\infty} V(\mathcal{E}_t) \leqslant \frac{\epsilon_{\text{sub}}}{\rho}. \tag{D.47}$$

This confirms that a suboptimal policy (e.g., the distilled student) simply adds a constant offset $\epsilon_{\text{sub}}/\rho$ to the final error bound, preserving stability as long as the distillation gap ϵ_{sub} is bounded. □

Theorem D.5 shows UUB with bound $\epsilon_{\text{sub}}/\rho$ under suboptimal policies. The following proposition confirms that ϵ_{sub} from auxiliary objectives (CBF, ADT, entropy) is bounded.

Proposition D.8 (Bounded Suboptimality Gap)**.** *The policy π_θ minimizes a composite multi-step objective:*

$$\mathcal{J}(\mathbf{A}) = \sum_{t=0}^{H-1} \gamma^t \left(\mathcal{L}_{CLF}^{(t)} + \mathcal{L}_{aux}^{(t)}\right), \tag{D.48}$$

where $\mathcal{L}_{aux}^{(t)}$ (by Definition B.15) includes CBF, ADT, and entropy terms. Let $\mathbf{A}^*$ minimize pure $V(\mathcal{E}_{t+1})$, and $\tilde{\mathbf{A}}$ minimize $\mathcal{J}$. The gap $\epsilon_{sub} = V\left(\mathcal{T}\left(\mathbf{S}, \tilde{\mathbf{A}}\right)\right) - V\left(\mathcal{T}\left(\mathbf{S}, \mathbf{A}^*\right)\right)$ is bounded by

$$\epsilon_{sub} \leqslant \sum_{k\in\{CBF,\ ADT,\ ent\}} \lambda_k C_k < \infty, \tag{D.49}$$

where C_k represents the theoretical upper bound of the respective loss term.

Proof. By optimality, we have:

$$\mathcal{L}_{\text{CLF}}\left(\tilde{\mathbf{A}}\right) + \mathcal{L}_{\text{aux}}\left(\tilde{\mathbf{A}}\right) \leqslant \mathcal{L}_{\text{CLF}}\left(\mathbf{A}^*\right) + \mathcal{L}_{\text{aux}}\left(\mathbf{A}^*\right);$$
$$\mathcal{L}_{\text{CLF}}\left(\tilde{\mathbf{A}}\right) - \mathcal{L}_{\text{CLF}}\left(\mathbf{A}^*\right) \leqslant \mathcal{L}_{\text{aux}}\left(\mathbf{A}^*\right) - \mathcal{L}_{\text{aux}}\left(\tilde{\mathbf{A}}\right).$$

Using $\mathcal{L}_{\text{CLF}} \approx \Delta V$, we have:

$$\epsilon_{\text{sub}} \leqslant \mathcal{L}_{\text{aux}}\left(\mathbf{A}^*\right) - \mathcal{L}_{\text{aux}}\left(\tilde{\mathbf{A}}\right).$$

Analyze the bounds of components in $\mathcal{L}_{\text{aux}}$:

- **CBF:** The barrier function involves bounded pressure states $\Phi \in [\epsilon, 1-\epsilon]^3$ (Proposition A.3), thus $\mathcal{L}_{\text{CBF}} \leqslant C_{\text{CBF}} < \infty$;
- **ADT:** The action dissimilarity is an indicator function or bounded distance, $\mathcal{L}_{\text{ADT}} \in [0, 1]$;
- **Entropy:** For a discrete action space $|\mathcal{A}|$, the entropy is bounded by $0 \leqslant \mathcal{H}(\pi) \leqslant \log|\mathcal{A}|$.

Since all auxiliary terms are strictly bounded and the regularization coefficients λ are finite constants, the RHS is bounded by a constant C_{total}. Therefore, $\epsilon_{\text{sub}} \leqslant C_{\text{total}} < \infty$. The condition for Theorem D.5 is satisfied, guaranteeing Ultimate Uniform Boundedness (UUB) with suboptimality under the multi-step optimized policy. □

Strategy	Weight	Priority	Aggression	Purpose
π_{bal}	40%	$\Phi > \mathcal{E} > \mathbf{H}$	Medium	Balanced decisions
π_{qual}	30%	$\mathcal{E} \gg \Phi$	High	Quality-focused data
π_{res}	20%	$\Phi \gg \mathcal{E}$	Low	Conservative data
π_{stab}	7%	$\mathbf{H} \gg \Phi$	Medium	Context-aware data
π_{danger}	3%	Counter-intuitive	Extreme	Negative samples

Table 4: Comparison of heuristic strategies. **Weight**: sampling probability; **Priority**: relative importance of state components ($\gg$ denotes "much greater than"); **Aggression**: retrieval intensity level.

E Heuristic Strategy: Definitions & Algorithms

This appendix provides complete specifications of 5 heuristic strategies we used in data generation for the training of the PAEC dynamics model $\mathcal{T}_\theta$.

E.1 Overview

The heuristic strategies operate on the system state $\mathbf{S}_t = [\mathcal{E}_t, \Phi_t, \mathbf{H}_t]^\top \in \mathbb{R}^{11}$ (Appendix A), comprising error (Definition A.1), pressure (Definition A.6), and context (Definition A.7) components. Each strategy outputs an action $\mathbf{A}_t = (\text{IndexType}, k, \lambda)$ per Definition B.3.

The strategies are designed to cover diverse regions of the state-action space, ensuring that the dynamics model learns from varied decision patterns.

Throughout this appendix, $\lambda_{\max} = 0.8$ bounds the interpolation weight to prevent over-reliance on retrieval, while $k_{\max} = 16$ bounds the neighbor count (Definition B.3).

Definition E.1 (Effective Confidence Penalty)**.** *To modulate retrieval intensity based on the NMT model's internal stability, we define an* effective confidence *score combining trajectory stability and historical volatility (cf. Definition A.7):*

$$C_{\mathit{eff}} = 0.7 \cdot H_t^{(\mathit{stab})} + 0.3 \cdot \left(1 - H_t^{(\mathit{vol})}\right)$$

where $H_t^{(\mathit{stab})} \in [0, 1]$ measures trajectory coherence and $H_t^{(\mathit{vol})} \in [0, 1]$ captures confidence fluctuation. The interpolation weight is penalized as:

$$\lambda_{\mathit{adj}} = \max\left\{0, \min\left\{\lambda \cdot \left(1 - C_{\mathit{eff}}\right), \lambda_{\max}\right\}\right\}$$

This penalty reduces retrieval reliance when the NMT model exhibits stable, confident behavior.

Definition E.2 (Heuristic Strategy Set)**.**

$$\Pi_{\mathit{heuristic}} = \{\pi_{\mathit{bal}}, \pi_{\mathit{qual}}, \pi_{\mathit{res}}, \pi_{\mathit{stab}}, \pi_{\mathit{danger}}\}$$

with sampling probabilities $(0.40, 0.30, 0.20, 0.07, 0.03)$.

Definition E.3 (Strategy Mixing)**.** *At each decoding step t, a strategy is independently sampled:*

$$\pi_t \sim \mathit{Categorical}\left([p_{\mathit{bal}}, p_{\mathit{qual}}, p_{\mathit{res}}, p_{\mathit{stab}}, p_{\mathit{danger}}]\right)$$

enabling fine-grained exploration of the state-action space.

E.2 Strategy Characteristics Summary

Table 4 summarizes the diversity of strategies, which ensures the training data covers:

(1). **Normal Operating** regions via π_{bal};
(2). **Quality-Critical** scenarios via π_{qual};
(3). **Resource-Constrained** scenarios via π_{res};
(4). **Model Instability** scenarios via π_{stab}; and
(5). **Dangerous Failure** Zones via π_{danger}.

This mixture is essential for the dynamics model $\mathcal{T}_\theta$ (Appendix B) to learn robust state-action mappings across the entire operational envelope.

E.3 π_{bal}: Balanced Policy

The balanced policy attempts to dynamically balance resources, quality, and context through hierarchical decision-making.

Priority: $\Phi > \mathcal{E} > \mathbf{H}$

Algorithm 3 Policy_Default_Balanced

Require: State components $\mathcal{E}_t, \Phi_t, \mathbf{H}_t$
Ensure: Action $\mathbf{A}_t = (\text{IndexType}, k, \lambda)$

Phase 1: Pressure Assessment
if $\phi_t^{\text{mem}} \geqslant 0.9$ **then**
  return $(\text{'none'}, 0, 0.0)$ ▷ Memory critical
else if $0.75 \leqslant \phi_t^{\text{mem}} < 0.9$ **then**
  IndexType $\leftarrow$ 'ivf_pq'
  $k \leftarrow \lfloor 4 - 3\left(\phi_t^{\text{mem}} - 0.75\right)/0.15 \rfloor$
else if $\phi_t^{\text{lat}} < 0.4 \wedge \phi_t^{\text{thr}} < 0.4$ **then**
  IndexType $\leftarrow$ 'exact'; $k \leftarrow 8$
else if $\phi_t^{\text{lat}} > 0.65$ **then**
  IndexType $\leftarrow$ 'hnsw'; $k \leftarrow 8$
else
  IndexType $\leftarrow$ 'hnsw'; $k \leftarrow 8$
end if
Phase 2: Error-Based Adjustment
if $\epsilon_t^{\text{sem}} > 0.8 \vee \epsilon_t^{\text{cov}} > 0.9$ **then**
  $(k, \lambda) \leftarrow (16, 0.5)$ ▷ Severe semantic/coverage error
else if $\epsilon_t^{\text{rep}} > 0.7$ **then**
  $(k, \lambda) \leftarrow (12, 0.4)$ ▷ Repetition issue
else if $\epsilon_t^{\text{surp}} > 0.8$ **then**
  $(k, \lambda) \leftarrow (8, 0.3)$ ▷ Fluency issue
else if $0.4\epsilon_t^{\text{sem}} + 0.3\epsilon_t^{\text{cov}} + 0.2\epsilon_t^{\text{surp}} + 0.1\epsilon_t^{\text{rep}} > 0.5$ **then**
  $(k, \lambda) \leftarrow (8, 0.33)$
else
  $(k, \lambda) \leftarrow (4, 0.2)$ ▷ Low error
end if
Phase 3: Confidence Penalty (cf. Definition E.1)
$C_{\text{eff}} \leftarrow 0.7H_t^{\text{stab}} + 0.3 \cdot \left(1 - H_t^{\text{vol}}\right)$
$\lambda_{\text{adj}} \leftarrow \max\left\{0, \min\left\{\lambda \cdot \left(1 - C_{\text{eff}}\right), \lambda_{\max}\right\}\right\}$
return $\left(\text{IndexType}, k, \lambda_{\text{adj}}\right)$

E.4 π_{qual}: Quality-First Policy

Aggressively pursues low error at the cost of resources. **Priority:** $\mathcal{E} \gg \Phi > \mathbf{H}$

Algorithm 4 Policy_Quality_First

Require: State components $\mathcal{E}_t, \Phi_t, \mathbf{H}_t$
Ensure: Action $\mathbf{A}_t$

Phase 1: Aggressive Error Correction
$\epsilon_{\text{weighted}} \leftarrow 0.5\epsilon_t^{\text{sem}} + 0.3\epsilon_t^{\text{cov}} + 0.1\epsilon_t^{\text{surp}} + 0.1\epsilon_t^{\text{rep}}$
if $\epsilon_{\text{weighted}} > 0.7$ **then**
  $(\text{IndexType}, k, \lambda) \leftarrow (\text{'exact'}, 16, 0.7)$
else if $\epsilon_{\text{weighted}} > 0.4$ **then**
  $(\text{IndexType}, k, \lambda) \leftarrow (\text{'hnsw'}, 12, 0.5)$
else if $\epsilon_{\text{weighted}} > 0.2$ **then**
  $(\text{IndexType}, k, \lambda) \leftarrow (\text{'hnsw'}, 8, 0.35)$
else
  $(\text{IndexType}, k, \lambda) \leftarrow (\text{'ivf\_pq'}, 4, 0.2)$
end if
Phase 2: Pressure Safety Valve
if $\phi_t^{\text{mem}} > 0.95$ **then**
  return $(\text{'none'}, 0, 0.0)$
else if $\phi_t^{\text{mem}} > 0.8$ **then**
  IndexType $\leftarrow$ 'ivf_pq'
  $k \leftarrow \max\left(1, \min\left(k, 4\right)\right)$
end if
Phase 3: Confidence Penalty (cf. Definition E.1)
$C_{\text{eff}} \leftarrow 0.7H_t^{\text{stab}} + 0.3 \cdot \left(1 - H_t^{\text{vol}}\right)$
$\lambda_{\text{adj}} \leftarrow \max\left\{0, \min\left\{\lambda \cdot \left(1 - C_{\text{eff}}\right), \lambda_{\max}\right\}\right\}$
return $\left(\text{IndexType}, k, \lambda_{\text{adj}}\right)$

E.5 π_{res}: Resource Guardian Policy

This extremely conservative policy allows minimal intervention only when resources are abundant and errors are severe. **Priority:** $\Phi \gg \mathcal{E} > \mathbf{H}$

Algorithm 5 Policy_Resource_Guardian

Require: State components $\mathcal{E}_t, \Phi_t, \mathbf{H}_t$
Ensure: Action $\mathbf{A}_t$

Phase 1: Pressure Threshold
if $\phi_t^{\text{mem}} > 0.6 \vee \phi_t^{\text{lat}} > 0.6 \vee \phi_t^{\text{thr}} > 0.6$ **then**
  return $(\text{'none'}, 0, 0.0)$ ▷ Pressure too high
end if
Phase 2: Error Threshold
$\epsilon_{\text{weighted}} \leftarrow 0.5\epsilon_t^{\text{sem}} + 0.4\epsilon_t^{\text{cov}} + 0.1\epsilon_t^{\text{rep}}$
if $\epsilon_{\text{weighted}} \leqslant 0.8$ **then**
  return $(\text{'none'}, 0, 0.0)$ ▷ Error not severe
end if
Phase 3: kNN Support Verification
if $H_t^{\text{con}} < 0.3$ **then**
  return $(\text{'none'}, 0, 0.0)$ ▷ Low kNN support quality
end if
return $(\text{'ivf\_pq'}, 2, 0.25)$ ▷ Min intervene

E.6 π_{stab}: Stability-Averse Policy[1]

Primarily concerned with stabilizing the decoder's internal state. **Priority:** $\mathbf{H} \gg \Phi > \mathcal{E}$

Algorithm 6 Policy_Stability_Averse

Require: State components $\mathcal{E}_t, \Phi_t, \mathbf{H}_t$
Ensure: Action $\mathbf{A}_t$

Phase 1: Cognitive Risk Assessment

$\gamma_{\text{cog}} \leftarrow 0.4 \cdot \left(1 - H_t^{\text{stab}}\right) + 0.3 H_t^{\text{vol}} + 0.2 \cdot \left(1 - H_t^{\text{foc}}\right) + 0.1 \cdot (1 - H_t^{\text{con}})$
if $\gamma_{\text{cog}} < 0.3$ **then**
  return ('none', 0, 0.0) ▷ Model stable
end if

Phase 2: Resource-Constrained Tool Selection

if $\phi_t^{\text{mem}} > 0.75$ **then** ▷ Choose settings
  $\mathbf{C}_{\text{tmp}} \leftarrow$ ('ivf_pq', 6, 0.35, 8)
else
  $\mathbf{C}_{\text{tmp}} \leftarrow$ ('hnsw', 10, 0.45, 16)
end if

Phase 3: Error-Based Intensity Tuning

$(\text{IndexType}, k, \lambda, k_{\max}) \leftarrow \mathbf{C}_{\text{tmp}}$
$\bar{\epsilon}_t \leftarrow \left(\epsilon_t^{\text{sem}} + \epsilon_t^{\text{cov}} + \epsilon_t^{\text{surp}} + \epsilon_t^{\text{rep}}\right)/4$
$k \leftarrow \max\left(1, \min\left(\text{round}\left(k\left(1 + 0.5\bar{\epsilon}_t\right)\right), k_{\max}\right)\right)$
$\lambda \leftarrow \lambda + 0.2\bar{\epsilon}_t$

Phase 4: Confidence Penalty (cf. Definition E.1)

$C_{\text{eff}} \leftarrow 0.7 H_t^{\text{stab}} + 0.3 \cdot \left(1 - H_t^{\text{vol}}\right)$
$\lambda_{\text{adj}} \leftarrow \max\left\{0, \min\left\{\lambda \cdot \left(1 - C_{\text{eff}}\right), \lambda_{\max}\right\}\right\}$
return $\left(\text{IndexType}, k, \lambda_{\text{adj}}\right)$

E.7 π_{danger}: Dangerous Perturbator

Deliberately makes counter-intuitive decisions under specific boundary conditions to generate negative samples.

Purpose: Teaching the dynamics model the consequences of "wrong" decisions.

Algorithm 7 Policy_Dangerous_Perturbator

Require: State components $\mathcal{E}_t, \Phi_t, \mathbf{H}_t$
Ensure: Action $\mathbf{A}_t$

Rule 1: Memory Pressure Sabotage

if $\phi_t^{\text{mem}} > 0.7$ **then**
  return ('exact', 16, 0.8)
end if

Rule 2: Severe Error Ignored

if $\epsilon_t^{\text{sem}} > 0.75 \vee \epsilon_t^{\text{cov}} > 0.9$ **then**
  return ('none', 0, 0.0)
end if

Rule 3: High-Quality External Knowledge Ignored

if $H_t^{\text{stab}} < 0.2 \wedge H_t^{\text{con}} > 0.8$ **then**
  return ('none', 0, 0.0)
end if

Rule 4: Resource Waste in Perfect State

if $\max\left(\phi_t^{\text{lat}}, \phi_t^{\text{mem}}, \phi_t^{\text{thr}}\right) < 0.2$ **then**
  if $\max\left(\epsilon_t^{\text{sem}}, \epsilon_t^{\text{cov}}, \epsilon_t^{\text{surp}}, \epsilon_t^{\text{rep}}\right) < 0.1$ **then**
    if $H_t^{\text{foc}} > 0.5 \wedge H_t^{\text{con}} > 0.8 \wedge H_t^{\text{stab}} > 0.8 \wedge H_t^{\text{vol}} < 0.1$ **then**
      return ('exact', 16, 0.9)
    end if
  end if
end if

Rule 5: Latency/Throughput Pressure

if $\phi_t^{\text{lat}} > 0.7 \vee \phi_t^{\text{thr}} > 0.7$ **then**
  return ('exact', 16, 0.8)
end if

Fallback: Maximum Aggression

return ('exact', 16, 0.9)

Remark E.1 (Connection to Dynamics Model Training)**.** *The trajectory data generated by these heuristic strategies serves as the training corpus for the dynamics model $\mathcal{T}_\theta$ (Definition B.1). The diverse "decision personalities" ensure coverage of:*

(i) ***State-space exploration****: Different priority orderings $\left(\Phi > \mathcal{E} > \mathbf{H}\right.$ vs. $\left.\mathcal{E} \gg \Phi$, etc.$\right)$ visit distinct regions of $\mathcal{S}$.*

(ii) ***Action-space diversity****: From aggressive retrieval (π_{qual}) to minimal intervention (π_{res}), the model observes varied control inputs.*

(iii) ***Failure modes****: The dangerous perturbator π_{danger} provides critical negative samples, teaching $\mathcal{T}_\theta$ the consequences of suboptimal decisions—essential for learning robust CLF-satisfying policies (cf. Definition D.5).*

This data generation strategy follows the principle of learning from diverse expert demonstrations*: distinct "decision personalities" contribute complementary state-action coverage, enabling the subsequent Lyapunov-guided training (Section 3.4) to inherit broad operational knowledge.*

[1] The name "Stability-Averse" reflects the policy's aversion to *instability*—i.e., it actively intervenes when the decoder exhibits unstable behavior.

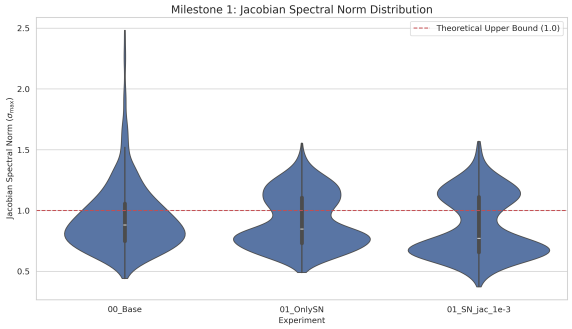

Figure 2: Jacobian spectral norm distributions. Red dashed line: theoretical bound (1.0). `00_Base` exhibits wide variance with tail $\approx$ 2.5; spectral normalization compresses the distribution to 0.5–1.5; adding Jacobian regularization further narrows it towards the level below 1.0.

Model	RMSE	Δ vs. baseline
`01_OnlySN`	0.182	—
`02_Text_Decoder`	0.161	-11.5%
`02_Predict_Delta`	0.121	-24.8%
`02_MultiHead`	0.124	-23.6% ($+2.5\%$ vs. Delta)

Table 5: RMSE across architectural variants. Text embeddings and delta prediction yield major improvements; multi-head adds slight overhead.

F Ablation Study Details

We organize ablation experiments through four progressive milestones, systematically validating theoretical predictions on the dynamics model $\mathcal{T}_\theta$. Each milestone addresses a specific theoretical requirement: Lipschitz continuity (robustness), prediction accuracy (fidelity), Lyapunov stability (convergence), and curriculum learning (optimization).

F.1 Milestone 1: Lipschitz Continuity

Lipschitz continuity (Definition C.1) bounds the rate of change of the dynamics model. This ensures small perturbations in state or action do not cause drastic changes in predictions—a prerequisite for closed-loop stability (cf. Appendix C).

F.1.1 Experimental Setup

We compare three configurations:

- `00_Base`: Unconstrained Transformer (no regularization);
- `01_OnlySN`: Spectral normalization on all linear layers;
- `01_SN_jac_1e-3`: Spectral normalization + Jacobian regularization $\left(\lambda_{\text{jac}} = 10^{-3}\right)$.

Figure 2 shows the Jacobian spectral norm distributions computed over the validation set using $\sigma_{\max}\left(\partial \mathcal{T}_\theta / \partial \mathbf{S}\right)$.

F.1.2 Results and Design Decision

The unconstrained `00_Base` exhibits wide variance with tail values approaching 2.5, confirming that vanilla Transformers lack inherent stability guarantees. Spectral normalization (`01_OnlySN`) effectively compresses the distribution to 0.5–1.5, with the median near 0.85. Adding Jacobian regularization (`01_SN_jac_1e-3`) further narrows the distribution to a level below 1.0.

Design decision: We adopt spectral normalization *without* Jacobian regularization as the standard configuration. While the latter provides marginally tighter bounds, it incurs excessive computational overhead without significant improvement in downstream stability metrics. Spectral normalization alone achieves $\mathbb{E}\left[\sigma_{\max}\right] \approx 1.02$ with 99th percentile ≈ 1.09, confirming near-non-expansive dynamics.

F.2 Milestone 2: Prediction Accuracy

Accurate dynamics prediction is necessary (but not sufficient) for stable control. This milestone evaluates architectural enhancements for predicting next-state error components $\hat{\mathcal{E}}_{t+1}$.

F.2.1 Architectural Variants

We progressively introduce three enhancements:

1. **Text Decoder** (`02_Text_Decoder`): Adds source/prefix embeddings and decoder hidden states as auxiliary inputs $\left(\text{cf. } \mathbf{I}_{\text{aux}} \text{ in Definition B.2}\right)$, providing semantic context for state prediction;
2. **Delta Prediction** (`02_Predict_Delta`): Instead of predicting absolute states $\hat{\mathcal{E}}_{t+1}$, it predicts state *increments* $\Delta\mathcal{E} = \mathcal{E}_{t+1} - \mathcal{E}_t$. This leverages the insight that learning *changes* is easier than memorizing absolute values;
3. **Multi-Head Output** (`02_MultiHead`): Separates prediction heads for error components $\mathcal{E}$ and context components $\mathbf{H}$, allowing for specialized representations.

F.2.2 Results

Table 5 summarizes RMSE improvements. **Key findings**:

(1). Text embeddings reduce RMSE by 11.5% by providing semantic grounding—the model can condition predictions on source content rather than relying solely on error signals;
(2). Delta prediction yields 24.8% improvement, confirming that learning state *changes* is fundamentally easier than absolute state prediction;
(3). Multi-head output shows slight regression (+2.5% vs. Delta), likely due to reduced parameter sharing; however, it provides cleaner separation for subsequent stability losses and is retained for interpretability.

F.3 Milestone 3: Lyapunov Stability

This milestone introduces explicit stability constraints via the Control Lyapunov Function (CLF) loss, as defined in Definition D.5. The Lyapunov function $V(\mathcal{E}) = \mathcal{E}^\top \mathbf{P} \mathcal{E}$ (Definition D.4) quantifies the error "energy" of the system. The key insight is that **accurate prediction does not imply stable decisions**—a model may predict dynamics correctly yet fail to guide the system toward low-error states.

F.3.1 Stability Loss Variants

We evaluate several stability loss formulations:

- `03_NCLF_1.0`: N-step CLF loss (Definition B.13) with decay rate $\rho = 1.0$
- `03_CVaR_0.7`: CVaR risk-averse variant ($\alpha = 0.7$) (Proposition D.5);
- `03_ADT_L1.3`: Action Dissimilarity Term (Definition B.15) with a Lipschitz bound 1.3;
- `04_No_Curriculum`: Full stability loss from epoch 0;
- `04_Curriculum_10`: Stability loss introduced at epoch 10 (Definition B.18).

F.3.2 Stability Metrics

We evaluate five complementary metrics:

1. **CLF Satisfaction Rate**: Probability that single-step predictions satisfy the CLF decreasing condition (Definition D.5): $V\left(\hat{\mathcal{E}}_{t+1}\right) \leqslant (1-\rho)\, V(\mathcal{E}_t)$;

Model	CLF	Drift	Cov.	Mono.	Viol.
`02_MultiHead`	59	28	7	0.089	—
`03_NCLF_1.0`	95	2	30	0.374	28
`03_CVaR_0.7`	97	1	37	0.416	27
`03_ADT_L1.3`	97	1	55	0.416	15
`04_No_Curr.`	> 99	< 1	54	0.425	15
`04_Curr._10`	> 99	< 1	43	**0.471**	**30**

Table 6: Final stability metrics (%). CLF: satisfaction rate (↑). Drift: positive drift rate (↓). Cov.: multi-step coverage (↑). Mono.: monotonic decay rate (↑). Viol.: N-step endpoint violation rate (↓).

2. **Positive Drift Rate**: Probability of energy *increase* ($\Delta V > 0$); lower is better;
3. **Multi-Step Coverage**: Proportion of steps in 20-step rollouts exhibiting negative drift;
4. **Monotonic Decay Rate**: Proportion of rollouts with *strictly* decreasing V at every step;
5. ***N*-Step Endpoint Violation Rate**: Proportion of rollouts where the final predicted state violates the CLF condition; lower is better.

F.3.3 Results: The Gap Between Prediction and Stability

Table 6 summarizes final metrics across model variants.

Critical finding: The pure prediction model (`02_MultiHead`) achieves only 59% CLF satisfaction with 28% positive drift and 7% multi-step coverage. Despite having the lowest RMSE (0.124), it produces *unstable* control decisions, confirming that accurate dynamics prediction is **necessary but not sufficient** for stability.

Introducing stability losses immediately transforms behavior: CLF satisfaction jumps to $> 95\%$, positive drift drops to $< 2\%$, and multi-step coverage increases 5–8×. This demonstrates the critical importance of explicit stability constraints.

F.3.4 Training Dynamics Analysis

Table 7 summarizes stability metrics at representative epochs from our S8 validation suite.

Three distinct behavioral patterns emerge from the training trajectories:

Pattern 1: Prediction-Only Degradation. `02_MultiHead` exhibits a counterintuitive *decline* in CLF satisfaction during training (68% → 59%). As the model overfits to minimize prediction

Model	Metric	Ep.1	Ep.5	Ep.10	Ep.15	Final
02_MH	CLF%	38	68	64	62	59
	Drift%	32	19	27	27	28
	Cov.%	7	9	8	7	7
	Viol.%	—	—	—	—	—
03_NCLF	CLF%	97	98	98	95	95
	Drift%	1	1	1	1	2
	Cov.%	29	30	31	30	30
	Viol.%	35	21	24	24	28
04_NoCurr.	CLF%	87	99	> 99	> 99	> 99
	Drift%	1	< 1	< 1	< 1	< 1
	Cov.%	25	44	47	55	54
	Viol.%	82	36	25	13	15
04_Curr.	CLF%	37	65	$63 \rightarrow 92$	99	> 99
	Drift%	35	22	$27 \rightarrow 4$	1	< 1
	Cov.%	6	8	$8 \rightarrow 28$	38	43
	Viol.%	98	99	$99 \rightarrow 70$	55	30

Table 7: Training dynamics at representative epochs. 02_MH: MultiHead (prediction only). 04_Curr.: Curriculum model shows phase transition at epoch 10 (marked by $\rightarrow$). Viol.%: N-step endpoint violation rate.

RMSE, it sacrifices stability properties. The positive drift rate initially drops ($32\% \rightarrow 19\%$) but then rebounds to 29%, while multi-step coverage stagnates at $\sim 7\%$. This confirms that optimizing prediction accuracy alone degrades stability—the model predicts dynamics accurately but fails to guide the system toward equilibrium.

Pattern 2: Rapid Stability Convergence. Models with stability loss from epoch 0 (03_NCLF_1.0, 04_No_Curriculum) achieve rapid convergence. 03_NCLF_1.0 reaches $> 97\%$ CLF satisfaction within 1 epoch, with drift stabilizing at $\sim$ 1–2%. However, 04_No_Curriculum shows an interesting trade-off: its N-step violation rate starts extremely high (82%) and requires $\sim$15 epochs to converge to 15%, revealing that joint optimization creates initial instability in long-horizon metrics despite strong single-step performance.

Pattern 3: Phase Transition in Curriculum Learning. 04_Curriculum_10 exhibits the most distinctive dynamics. During Phase 1 (epochs 1–10), it behaves similarly to 02_MultiHead: CLF satisfaction hovers at 55–67%, drift remains high (20–35%), and coverage stagnates at $\sim$8%. The N-step violation rate stays critically high at $\sim$99%—nearly all rollouts violate endpoint constraints.

At epoch 10, when stability loss is introduced, a dramatic **phase transition** occurs:

- CLF satisfaction: $63\% \rightarrow 92\%$ within one epoch, then $\rightarrow (> 99\%)$;
- Positive drift: $27\% \rightarrow 4\%$, then $\rightarrow (< 1\%)$;
- Multi-step coverage: $8\% \rightarrow 28\%$, eventually reaching 43%;
- N-step violation: $99\% \rightarrow 70\%$, converging to 30% (the transient spike reflects the model adapting to the new constraint space).

This phase transition demonstrates that curriculum learning enables the model to first establish an accurate understanding of dynamics and then rapidly acquire stability properties without catastrophic forgetting of prediction accuracy.

F.3.5 N-Step Endpoint Violation Analysis

The N-step endpoint violation rate measures the proportion of 20-step rollouts where the *final* predicted state violates the CLF condition—a direct measure of long-horizon planning failure. This metric reveals critical differences that are invisible to single-step metrics.

Key observations:

1. 04_No_Curriculum starts with 82% violation despite 87% single-step CLF satisfaction, revealing that strong local stability does not guarantee long-horizon convergence. Joint optimization from epoch 0 creates competing gradients that initially destabilize multi-step behavior;
2. 04_Curriculum_10 maintains $\sim$ 99% violation during Phase 1, then drops sharply. Its final violation rate (30%) is higher than 04_No_Curriculum (15%), but this is compensated by a superior monotonic decay rate (0.471 vs. 0.425)—the curriculum model produces fewer "perfect" rollouts but more consistent overall stability;
3. 03_ADT_L1.3 achieves the best balance: lowest violation rate (15%) among non-curriculum models while matching the top monotonic decay (0.416). The adversarial domain training explicitly optimizes for worst-case robustness, which translates to superior endpoint guarantees.

F.3.6 Monotonic Decay Comparison

Figure 3 compares final monotonic decay rates—the strictest stability metric requiring *every* step in a 20-step rollout to decrease V.

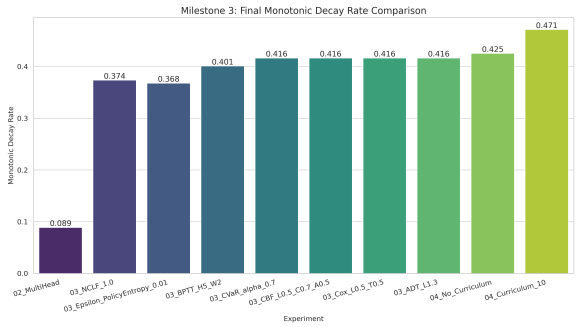

Figure 3: Final monotonic decay rate comparison across all Milestone 3 variants. `04_Curriculum_10` achieves the highest rate (0.471), indicating that 47.1% of rollouts exhibit strictly decreasing error energy at every step. The gap between `02_MultiHead` (0.089) and stability-constrained models (> 0.37) confirms the critical importance of explicit stability objectives.

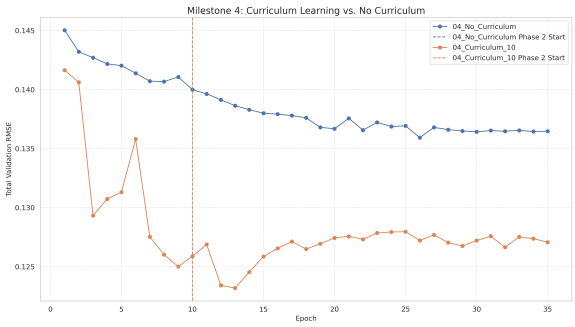

Figure 4: Curriculum learning comparison. `04_Curriculum_10` (orange) achieves lower final RMSE (0.127 vs. 0.137). Vertical dashed line marks Phase 2 introduction. Note the transient perturbation at epoch 10 as the model adapts to the new constraint.

The monotonic decay rate reveals a clear hierarchy:

- `02_MultiHead`: 0.089 (baseline, prediction-only);
- `03_NCLF_1.0`: 0.374 (4.2× improvement);
- `03_CVaR_0.7/ADT_L1.3`: 0.416 (risk-aware variants plateau);
- `04_No_Curriculum`: 0.425;
- `04_Curriculum_10`: **0.471** (best, 5.3× baseline).

`04_Curriculum_10` achieves the highest monotonic decay rate (0.471), outperforming even `04_No_Curriculum` (0.425) despite its delayed stability training. This suggests that the phased approach—allowing the model to first master dynamics prediction before introducing stability constraints—yields more robust long-horizon stability guarantees. The curriculum strategy finds a "smoother" region of the loss landscape where prediction & stability objectives are better aligned.

F.4 Milestone 4: Curriculum Learning

When simultaneously optimizing prediction ($\mathcal{L}_{\text{pred}}$) and stability ($\mathcal{L}_{\text{stab}}$), a fundamental challenge emerges: these objectives may conflict in the early stages of training. Premature stability constraints can interfere with learning basic dynamics, preventing either objective from being fully optimized. Hence, we implement two-phase curriculum learning (Definition B.18) to decouple conflicting training objectives.

F.4.1 Curriculum Strategy

We compare two approaches:

- `04_No_Curriculum`: Joint optimization from epoch 0;
- `04_Curriculum_10`: Phase 1 (epochs 1–10) optimizes only $\mathcal{L}_{\text{pred}}$; Phase 2 (epochs 11+) introduces $\mathcal{L}_{\text{stab}}$, following Definition B.18.

F.4.2 Results

Figure 4 compares RMSE trajectories.

Key observations:

1. `04_No_Curriculum` shows slower, noisier convergence (final RMSE 0.137), as multi-objective optimization creates competing gradients from the start;
2. `04_Curriculum_10` rapidly decreases RMSE during Phase 1 (pure prediction, epochs 1–10), reaching ~ 0.125 by epoch 10;
3. **Transient perturbation**: At epoch 10, introducing stability loss causes a brief RMSE spike as the model adapts to the new constraint space. However, recovery is rapid ($\sim$ 2–3 epochs);
4. Final convergence: `04_Curriculum_10` achieves RMSE 0.127 *and* monotonic decay 0.471, outperforming `04_No_Curriculum` on both metrics (0.137 RMSE, 0.425 monotonic decay).

This demonstrates that curriculum learning effectively **decouples** conflicting objectives: prioritizing exploration in early training helps escape local optima, while gradual stability introduction guides the optimizer toward Pareto-optimal solutions.

Model	RMSE	CLF	Mono.	Jac.
03_CVaR_0.7	0.134	96%	0.416	0.98
03_ADT_L1.3	0.131	97%	0.416	0.99
04_No_Curr.	0.137	$> 99\%$	0.425	0.97
04_Curr._10	**0.127**	$> 99\%$	**0.471**	0.98

Table 8: Finalist model comparison. Jac.: average Jacobian norm.

F.5 Champion Model Selection

Based on comprehensive evaluation across all milestones, we select the optimal configuration for production deployment.

F.5.1 Multi-Dimensional Comparison

Table 8 summarizes key metrics for finalist models.

F.5.2 Selection Rationale

04_Curriculum_10 emerges as the optimal choice:

1. **Best prediction accuracy**: Lowest RMSE (0.127), critical for accurate dynamics modeling;
2. **Highest monotonic decay**: 0.471 vs. 0.425 for the next best, indicating superior long-horizon stability;
3. **Top-tier CLF satisfaction**: $> 99\%$, matching the best stability-focused models;
4. **Controlled Lipschitz constant**: Average Jacobian norm 0.98, confirming near-non-expansive dynamics;
5. **Robustness**: The curriculum approach produces "smoother" solutions less prone to overfitting.

F.5.3 Champion Model Training

The final Champion model applies the 04_Curriculum_10 configuration to a larger dataset (strategy_comparison_stepwise_1000.csv, with $10\times$ more samples per strategy) for production deployment. This model serves as the dynamics core for both PAEC Online (gradient-based planning) and PAEC Offline (distilled policy network) variants evaluated in the main experiments.

G Extended Experimental Analysis

This appendix provides a detailed experimental analysis complementing the main results, including attribution analysis, risk-stratified evaluation, trajectory case studies, and a comprehensive discussion of limitations.

Throughout this appendix, we employ definitions from preceding appendices: error state $\mathcal{E}_t$ (Definition A.1), resource pressure Φ_t (Definition A.6), Lyapunov function $V(\mathcal{E}) = \mathcal{E}^\top \mathbf{P} \mathcal{E}$ (Definition D.4), and dynamics model $\mathcal{T}_\theta$ (Definition B.1).

G.1 Result Attribution: Datastore Scale vs. Control Strategy

The observed "safety tax" (PAEC's 1.4-point BLEU deficit vs. Adaptive kNN-MT) requires careful attribution. This gap reflects **distinct operating regimes**—active retrieval (PAEC) versus abandoned retrieval (Adaptive)—rather than inherent framework limitations.

G.1.1 The Asymmetry Problem

Our experimental setup creates significant asymmetry between the NMT model and retrieval components:

- **NMT Model**: Transformer-Base trained on millions of sentence pairs, representing a strong parametric component.
- **Datastore**: Only $50,000$ sentence pairs, constrained by a single A100 GPU memory when initializing three FAISS indices simultaneously.

This "strong model, weak retrieval" configuration amplifies behavioral differences between methods:

Adaptive kNN-MT: Degeneration Strategy. The Meta-k network learns to predict near-zero λ values when datastore quality is poor, effectively degenerating to Pure NMT. Quantitative evidence:

- BLEU scores nearly identical: Adaptive (36.68) $\approx$ Pure NMT (36.60);
- Average Lyapunov error: Adaptive $(0.385) \approx$ Pure NMT (0.383);
- Failure rates (proportion of samples with final $V(\mathcal{E}) > 0.5$): Adaptive $(28.82\%) \approx$ Pure NMT (28.63%).

Subset	Samples	Mean $V(\mathcal{E}_0)$	Std
Low-Risk	902	0.187	0.092
High-Risk	1,034	0.584	0.213

Table 9: Dataset partition by initial risk. Low-Risk: $\epsilon_0^{(\text{cov})}=0$ AND $V_0<0.8$; High-Risk: $\epsilon_0^{(\text{cov})}=1$ OR $V_0>1.2$. Excludes 64 intermediate samples (1,936 total).

This strategy preserves average performance by *abandoning retrieval* but forfeits external correction capability when the NMT model itself fails.

PAEC: Persistent Stabilization Strategy. The Lyapunov control law prioritizes trajectory stability over instantaneous probability. When pressure components satisfy $\max\left(\phi_t^{\text{lat}}, \phi_t^{\text{mem}}, \phi_t^{\text{thr}}\right) \leqslant 0.9$ (Definition A.6), PAEC *forces* retrieval to bound error accumulation, even when this introduces noise from the weak datastore.

G.1.2 Implications for Production Deployment

We hypothesize that at larger scales with higher-quality datastores:

1. Retrieval would provide genuine quality improvements rather than noise;
2. The observed BLEU trade-off would diminish as datastore quality increases;
3. PAEC's formal stability guarantees would become increasingly valuable as system complexity grows.

Validating these hypotheses requires production-scale experiments, which we identify as critical future work.

G.2 High-Risk Scenario Analysis

We partition test samples by initial Lyapunov energy $V(\mathcal{E}_0)$ (Definition D.4) into two subsets:

G.2.1 Rank Distribution Analysis

For each sample, we rank the final $V(\mathcal{E})$ across all five models (Rank 1 = best stability, Rank 5 = worst). Figure 5 shows the distribution for high-risk samples.

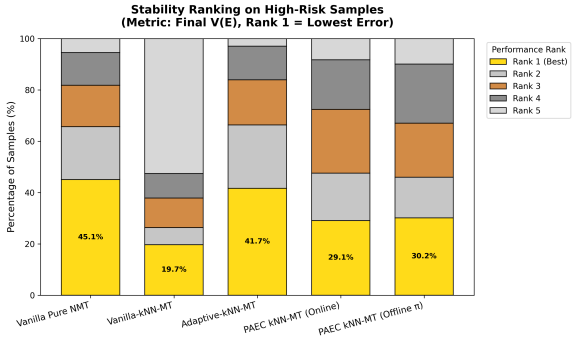

Figure 5: Stability rank distribution on high-risk samples (1,034 samples). Rank 1 (gold) = lowest final error. Pure NMT dominates Rank 1 (45.1%) due to our "strong model, weak datastore" setup, but PAEC demonstrates consistent Rank 4/5 avoidance.

Model	R1	R2	R3	R4	R5
Pure NMT	45.1	20.1	15.2	13.4	6.2
Vanilla	19.7	6.1	8.4	13.7	52.1
Adaptive	41.7	23.2	18.5	13.8	2.8
PAEC (Online)	29.1	16.4	19.2	27.0	8.3
PAEC (Offline)	30.2	14.7	22.1	24.1	8.9

Table 10: Full rank distribution (%) on high-risk samples. R1–R5 denote Rank 1 (best) through Rank 5 (worst).

G.2.2 Interpreting the Results

Dominance of Pure NMT. The 45.1% Rank 1 rate for Pure NMT appears counterintuitive but reflects our experimental constraints. In a "strong model, weak datastore" setup, "doing nothing" (Pure NMT) often outperforms "doing something wrong" (noisy retrieval). This is evident from Vanilla kNN-MT's catastrophic 52.1% Rank 5 rate—fixed retrieval from a weak datastore introduces harmful noise.

Adaptive's Trade-off. Adaptive achieves 41.7% Rank 1 through aggressive optimization, with only 2.8% Rank 5. However, this comes from degenerating to Pure NMT behavior, forfeiting retrieval's potential benefits.

PAEC's Safety Net. While PAEC Offline achieves lower Rank 1 (30.2%), it demonstrates **consistent Rank 4/5 avoidance** (combined 33.0% vs. Vanilla's 65.8%). This confirms PAEC's value: **minimizing catastrophic failures** rather than maximizing best-case outcomes.

G.2.3 Low-Risk vs. High-Risk Comparison

On low-risk samples (902), the pattern shifts:

- Pure NMT: 50.2% R1 (vs. 45.1% on high-risk);
- Adaptive: 45.5% R1 (vs. 41.7% on high-risk);
- PAEC Offline: 31.4% R1 (vs. 30.2% on high-risk).

The gap between PAEC and baselines *narrows* on high-risk samples, confirming PAEC's relative advantage in volatile scenarios where stability matters most.

G.3 Case Studies: Critical Divergence Scenarios

To demonstrate PAEC's stabilization mechanism across diverse high-risk scenarios, we analyze four representative samples where PAEC achieves significant gains over Adaptive kNN-MT. Figure 6 presents the trajectory comparisons, with green shaded regions indicating PAEC's stability advantage.

Sample	Steps	V_0	V_T^{Adap}	V_T^{PAEC}	ΔV
403	40	1.8	0.35	0.10	0.16
1260	35	1.5	0.35	0.10	0.28
1997	25	1.6	0.35	0.05	0.29
762	15	1.75	1.10	0.20	**0.94**

Table 11: Trajectory summary for four high-risk samples. V_0: initial error; V_T: final error (approx.); $\Delta V = V_T^{\text{Adap}} - V_T^{\text{PAEC}}$ in Figure 6.

G.3.1 Overview of Four Samples

Table 11 summarizes the trajectory characteristics across all four samples.

G.3.2 Sample 762: The Most Dramatic Divergence

Sample 762 (bottom-right) in Figure 6 exhibits the largest separation ($\Delta V = 0.94$), making it the clearest demonstration of PAEC's value.

Adaptive Failure Mode. The Adaptive trajectory (red dashed) shows:

1. **Steps 0–5**: Both methods decline similarly during early token processing;

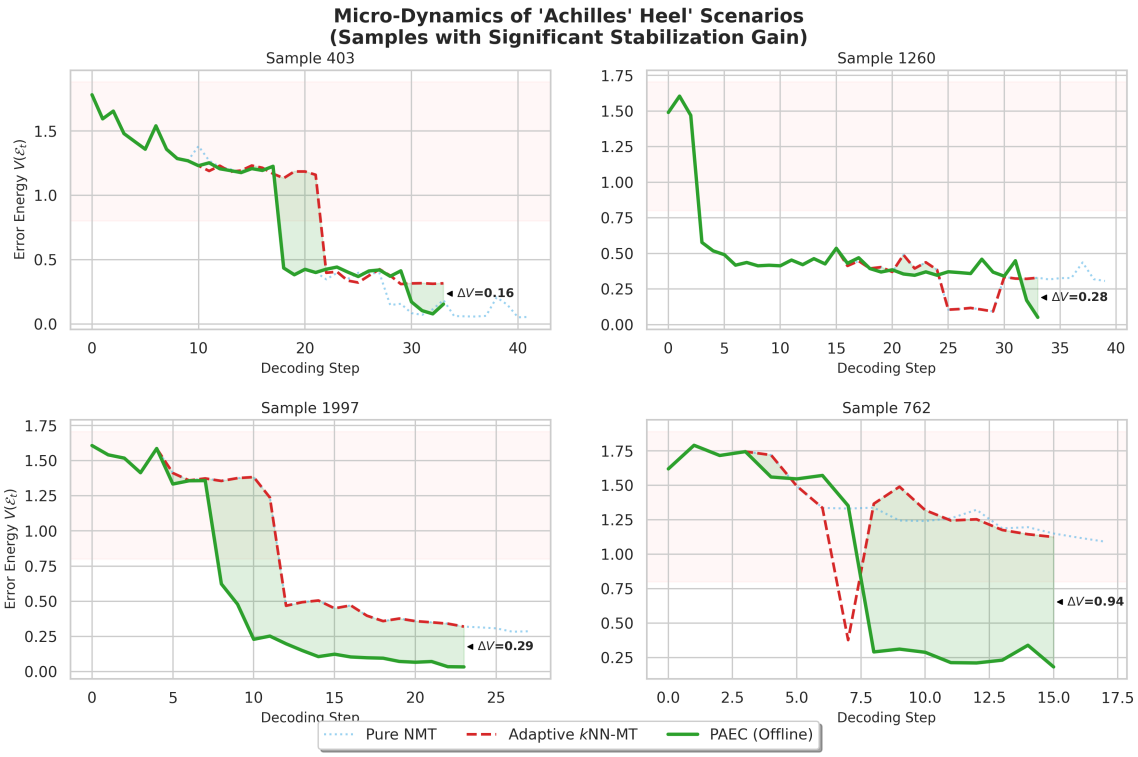

Figure 6: Micro-dynamics of four "Achilles' heel" scenarios. Each subplot compares Adaptive kNN-MT (red dashed), PAEC Offline (green solid), and Pure NMT (light blue dotted). Green shaded regions indicate PAEC's stability advantage. Sample 762 (bottom-right) exhibits the largest separation ($\Delta V = 0.94$), while the other three samples show consistent but smaller gains ($\Delta V = 0.16$–0.29).

2. **Steps 5–8**: Critical decision point—Adaptive's Meta-k network predicts moderate confidence, skipping retrieval despite impending divergence;
3. **Steps 8–15**: Error energy stabilizes at ~ 1.1–1.35; the system enters a "trapped" state where accumulated errors prevent recovery.

Note that Pure NMT (light blue dotted) follows a similar high-error trajectory, confirming that the base model itself struggles with this sample—the scenario where retrieval-based correction is most needed, yet Adaptive abandons it.

PAEC Intervention. The PAEC trajectory (green solid) demonstrates active stabilization:

1. **Steps 5–8**: Dynamics model $\mathcal{T}_\theta$ predicts divergence risk; Lyapunov-minimizing action selected:

$$\mathbf{A}^* = \arg\min_{\mathbf{A} \in \mathcal{A}} V\left(\mathcal{T}_\theta\left(\mathbf{S}, \mathbf{A}\right)_{\mathcal{E}}\right);$$

2. **Steps 8–15**: Forced intervention pulls trajectory into the stable region (green shaded), converging to $V \approx 0.25$—approximately $4\times$ lower than Adaptive's $V \approx 1.1$.

G.3.3 Samples 403, 1260, 1997: Consistent Moderate Gains

The remaining three samples reveal complementary patterns:

Sample 403 (Top-Left, ΔV=0.16). This sample in Figure 6 shows the longest trajectory (40 steps), exhibiting a **late-stage divergence pattern**:

- Steps 0–15: All three methods follow similar declining trajectories from $V \approx 1.8$ to $V \approx 1.1$;
- Steps 15–20: Critical separation—a transient spike at step 17–18 triggers divergence; PAEC responds with stabilizing intervention while Adaptive continues passively;
- Steps 20–40: PAEC converges to $V \approx 0.1$ while Adaptive stabilizes at $V \approx 0.3$.

This demonstrates PAEC's advantage in **long-horizon stability**: even when early behavior is similar, PAEC's trajectory-aware control prevents late-stage stagnation.

Sample 1260 (Top-Right, ΔV=0.28). This sample in Figure 6 shows **oscillatory behavior**:

- Adaptive (red) exhibits significant oscillations throughout decoding, with multiple local peaks;
- PAEC (green) shows smoother convergence with dampened oscillations;
- The gap emerges gradually, reaching $\Delta V = 0.28$ by step 35.

This highlights PAEC's **variance reduction**: the Lyapunov constraint not only reduces final error but also stabilizes the trajectory shape.

Sample 1997 (Bottom-Left, ΔV=0.29). This sample in Figure 6 demonstrates **early intervention effectiveness**:

- Steps 0–5: Initial high error (~ 1.6) with both methods declining
- Steps 5–10: Critical separation point—PAEC drops sharply while Adaptive plateaus
- Steps 10–25: PAEC converges to near-zero (~ 0.05), achieving the lowest absolute final error among all four samples

Notably, PAEC's final error (0.05) is lower than even the trajectory of Pure NMT, demonstrating that well-timed retrieval intervention can *outperform* the base model.

G.3.4 Cross-Sample Patterns

Comparing all four samples reveals systematic patterns:

1. **Separation timing varies**: Sample 762 diverges early (step 5–8), while Sample 403 diverges late (step 15–20). PAEC adapts to both patterns through its dynamics model.
2. **Adaptive tracks Pure NMT**: In all samples, Adaptive (red) closely follows Pure NMT (light blue), confirming that the Meta-k network predicts near-zero λ and effectively degenerates to parametric-only decoding.
3. **Green shading = opportunity cost**: The shaded regions quantify what Adaptive "leaves on the table" by abandoning retrieval. Larger shading indicates greater missed stabilization opportunity.
4. **Absolute vs. relative gains**: Sample 762 exhibits the largest relative improvement ($\Delta V = 0.94$), while Sample 1997 achieves the lowest absolute final error ($V = 0.05$). This demonstrates that PAEC's stabilization benefits are context-dependent.

G.3.5 Lessons Learned

These four samples collectively illustrate three fundamental principles:

1. **Greedy heuristics fail at decision boundaries**: Confidence-based thresholds create blind spots where moderate confidence masks impending divergence. The Meta-k network optimizes single-step probability, not trajectory stability;
2. **Lyapunov control provides lookahead**: By predicting multi-step consequences via $\mathcal{T}_\theta$ (Definition B.1), PAEC identifies divergence risk *before* it manifests in immediate metrics;
3. **Conservative actions pay off**: The short-term "cost" of forced retrieval is consistently outweighed by long-term stability gains, with ΔV ranging from 0.16 to 0.94 across diverse scenarios.

G.4 Comprehensive Limitations

We acknowledge several limitations that contextualize our findings with directions for future work.

G.4.1 Experimental Constraints

Datastore Scale. The 50K-sentence datastore is orders of magnitude smaller than production systems (billions of entries), constrained by single A100 GPU memory when initializing three FAISS indices. Simulation methods (e.g., synthetic latency injection) could bridge this gap. Future work will explore:

- Distributed FAISS indices across more GPUs;
- Simulation-based scaling analysis to predict production behavior;
- Collaboration with industry partners for production-scale evaluation.

Language Coverage. Experiments are confined to German-English (De-En). While the control-theoretic framework is language-agnostic, several aspects remain unverified:

- Multilingual generalization, especially for distant language pairs (e.g., En-Zh, En-Ja);
- Low-resource language performance where datastores may be inherently sparse;
- Morphologically rich languages where error metrics (especially repetition) may behave differently.

Domain Diversity. The `OPUS-100/Europarl` mixture emphasizes legal and multi-domain text. Performance on specialized domains (medical, technical, conversational) requires separate validation to ensure error metrics generalize.

G.4.2 Methodological Limitations

Baseline Scope. We compare against Vanilla and Adaptive kNN-MT to isolate Lyapunov control contributions. Notably, PAEC is *model-agnostic* and *orthogonal* to recent efficiency advances in kNN-MT:

- Token-level filtering for faster retrieval (Shi et al., 2024);
- Representation smoothing via `INK` (Zhu et al., 2023).

These works optimize the retrieval mechanism itself, whereas PAEC optimizes *decision dynamics*. Consequently, PAEC can be applied **on top** of these modern baselines to provide formal stability guarantees without conflict.

Lyapunov Function Design. The quadratic Lyapunov function (Definition D.4) with diagonal $\mathbf{P}$ assumes independent error components. Cross-component interactions exist in practice. Future work may explore:

- Non-diagonal $\mathbf{P}$ capturing error correlations;
- Neural Lyapunov functions (Wu et al., 2023) learned end-to-end;
- Adaptive weighting schemes based on context.

Offline Distillation. Behavioral cloning (Definition B.21) preserves the teacher's risk profile (29.3% failure rate) but may miss nuances of gradient-based planning. While Theorem D.5 guarantees bounded suboptimality, approaches to close the Online-Offline gap include:

- DAgger-style iterative refinement;
- Reward shaping to emphasize critical decision;
- Trajectory-level imitation learning.