AIR Journal of Mathematics & Computational Sciences, Vol. 2026
Article Unique ID: AIRMCS2026525
DOI: 10.65737/AIRMCS2026525

IN-CONTEXT LEARNING CHARACTERIZATION OF NAVIGATION AND GNSS FOUNDATION MODELS: A THEORETICAL FRAMEWORK FOR SAFETY-CRITICAL POSITIONING

Mosab Hawarey
Director, Geospatial Research (*director@geospatial.ch*)

Received: 2026 March 10 | Revised: 2026 March 14 | Approved: 2026 March 15 | Published: 2026 March 16

Abstract

Foundation models for navigation and Global Navigation Satellite Systems (GNSS) exhibit a striking empirical dichotomy: transformer-based architectures achieve remarkable performance on trajectory prediction and state estimation, yet consistently struggle with fault detection, cycle slip identification, and integrity monitoring. Despite the proliferation of navigation transformers, no theoretical framework explains which navigation tasks admit efficient in-context learning (ICL) and which are fundamentally intractable. This gap has critical implications for safety-critical applications including aviation, autonomous vehicles, and maritime navigation, where theoretical guarantees are essential for certification.

We address this gap by extending the ICL characterization framework to navigation and GNSS domains. Our main result is the **Navigation Dichotomy Theorem**: every natural navigation function class falls into exactly one of two categories. *Type A (ICL-Easy)* tasks—including position estimation, velocity estimation, atmospheric delay modeling, and trajectory prediction—possess additive sufficient statistics computable by attention mechanisms, achieving sample complexity matching the statistical optimum. *Type C (ICL-Hard)* tasks—including satellite fault detection, cycle slip detection, spoofing detection, and integrity monitoring—require combinatorial sufficient statistics over $C(M,J)$ fault hypotheses, rendering them intractable for standard ICL when the number of faults J exceeds the threshold $J^* \approx 2–3$ (for navigation; $J^* \approx 2–4$ across GeoAI domains).

Building on recent advances in chain-of-thought (CoT) reasoning, we prove that CoT prompting provides a mechanism to overcome ICL-Hard barriers in navigation. We establish that $T = \Omega(J)$ tokens are necessary and sufficient for J-fault detection, with a navigation-specific token multiplier $\kappa_{nav} \approx 3.5$. The effective hardness threshold shifts to $J^*_{CoT}(T) \approx J^* + T/\kappa_{nav}$, enabling detection of up to 7 simultaneous faults with 20 reasoning tokens.

We provide complete classification of 32 navigation tasks across six categories (Table 6), derive seven testable predictions for navigation transformer behavior, and analyze implications for safety certification under aviation (DO-229D/DO-253D), automotive (ISO 26262), and maritime (IMO) standards. Our results establish that hybrid architectures—combining ICL for state estimation with explicit algorithms for integrity monitoring—are not merely practical conveniences but fundamental computational necessities. This work completes a seven-paper theoretical program characterizing ICL across all major geospatial AI domains, establishing a

unified science of when and why foundation models succeed or fail on Earth observation and navigation tasks.

Keywords: in-context learning; navigation foundation models; GNSS; transformer architectures; computational complexity; fault detection; integrity monitoring; autonomous vehicles; aviation safety; chain-of-thought prompting

Citation: Hawarey, M. (2026). In-Context Learning Characterization of Navigation and GNSS Foundation Models: A Theoretical Framework for Safety-Critical Positioning. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026525, DOI: 10.65737/AIRMCS2026525.

1. Introduction

1.1 The Navigation AI Revolution

The convergence of deep learning and navigation has initiated a fundamental transformation in positioning, navigation, and timing (PNT) systems. Foundation models—large-scale neural networks trained on diverse datasets and capable of adapting to new tasks through in-context learning—are increasingly being applied to navigation challenges that have traditionally relied on model-based algorithms. This shift promises more flexible, adaptive navigation systems capable of operating in environments where classical approaches struggle.

The emergence of transformer-based navigation models represents a particularly significant development. TrajectoryTransformer (Janner et al., 2022) demonstrated that transformer architectures could achieve state-of-the-art performance in trajectory prediction by treating navigation as sequence modeling. Subsequent work has applied attention mechanisms directly to raw pseudorange and carrier phase observations, learning to estimate position without explicit geometric models. Multi-sensor fusion approaches combine GNSS, inertial, and visual inputs through cross-attention mechanisms, while integrity-focused architectures attempt to apply transformers to fault detection and protection level estimation. Although comprehensive surveys of these emerging architectures remain sparse, the trajectory prediction literature—including AgentFormer (Yuan et al., 2021), Trajectron++ (Salzmann et al., 2020), and Social LSTM/GAN variants (Alahi et al., 2016; Gupta et al., 2018)—demonstrates the broader trend of transformer adoption in navigation-adjacent domains.

These developments are driven by the remarkable success of in-context learning (ICL) in natural language processing and computer vision, building on the transformer architecture (Vaswani et al., 2017) and the demonstration that large language models can perform few-shot learning through in-context examples (Brown et al., 2020). ICL enables models to perform new tasks by conditioning on a few examples provided in the input context, without updating model parameters. For navigation, this capability suggests the possibility of receivers that adapt to new environments, handle novel error sources, and generalize across different satellite constellations—all through the examples presented in their context rather than through explicit reprogramming.

1.2 The Empirical Puzzle

Despite the enthusiasm surrounding navigation foundation models, a puzzling empirical pattern has emerged. These models exhibit dramatically different performance across navigation tasks, with no clear explanation for the disparity.

On continuous estimation tasks, navigation transformers perform remarkably well. Position estimation from pseudoranges achieves accuracy comparable to classical least-squares solutions. Velocity estimation from Doppler measurements shows similar success. Trajectory prediction—forecasting future vehicle positions from historical observations—has become a showcase application, with transformer models achieving state-of-the-art results on benchmarks such as nuScenes (Caesar et al., 2020), Argoverse (Chang et al., 2019), and Waymo Open Dataset (Sun et al., 2020). Atmospheric delay modeling, particularly for ionospheric corrections, benefits from the spatial interpolation capabilities of attention mechanisms.

In stark contrast, discrete detection tasks remain stubbornly difficult. Satellite fault detection—identifying which of the visible satellites are providing erroneous measurements—shows poor performance when more than one or two faults are present simultaneously. Cycle slip detection, the task of identifying integer discontinuities in carrier phase measurements, fails when multiple slips occur across different satellites or time epochs. Spoofing detection, critical for security applications, proves unreliable against sophisticated attacks involving multiple counterfeit signals. Most concerningly, integrity monitoring—the computation of protection levels that bound position error with extremely high confidence—shows large gaps between transformer-based approaches and traditional Receiver Autonomous Integrity Monitoring (RAIM) algorithms.

This pattern is not unique to any particular architecture or training regime. Larger models, more training data, and sophisticated augmentation strategies improve performance on the easy tasks but fail to close the gap on the hard ones. The dichotomy appears fundamental rather than incidental.

1.3 The ICL Characterization Framework

This paper explains the empirical puzzle through the theoretical lens of in-context learning characterization. We build upon a framework developed in a series of companion papers (Hawarey, 2026a–f) that establishes precise conditions determining which tasks admit efficient ICL and which are fundamentally intractable.

The foundation of this framework is *Sufficient Statistic Complexity (SSC)*, introduced in Paper 1 (Hawarey, 2026a). For a function class $\mathscr{F}$, the SSC measures the minimal dimension of statistics that are sufficient for learning: $SSC(\mathscr{F}) = min\{dim(S) : S \text{ is sufficient for } \mathscr{F}\}$. The crucial refinement is *Attention-Computable SSC (AC-SSC)*, which requires the sufficient statistic to be computable by attention mechanisms—specifically, to admit an additive decomposition $S = \Sigma_i \varphi(x_i, y_i)$ where φ is computable by a single attention layer.

Paper 1 established the *Master Dichotomy Theorem*: every natural function class is either *ICL-Easy* (polynomial AC-SSC, optimal sample complexity) or *ICL-Hard* (exponential AC-SSC, super-polynomial resource requirements). The proof connects transformer computation to circuit complexity, showing that constant-depth polynomial-size transformers compute functions in TC^0. Håstad's celebrated lower bounds then imply that certain tasks—specifically those encoding sparse parity—require either super-constant depth or super-polynomial size.

This framework has been successfully applied across geospatial AI domains. Paper 2 (Hawarey, 2026b) characterized quantum-enhanced geopotential estimation, showing that smooth field estimation is ICL-Easy while discrete source localization is ICL-Hard. Paper 3 (Hawarey, 2026c) established the Remote Sensing Dichotomy: pixel-wise tasks (classification, segmentation) are ICL-Easy while instance-level tasks (object detection, counting) are ICL-Hard for more than $J^* \approx 2–4$ objects. Paper 4 (Hawarey, 2026d) extended this to climate foundation models, and Paper 5 (Hawarey, 2026e) to multi-modal geo-foundation models.

Paper 6 (Hawarey, 2026f) made a crucial advance by showing that chain-of-thought (CoT) prompting provides a mechanism to overcome ICL-Hard barriers. The *CoT Depth Amplification Theorem* proves that generating T intermediate tokens provides effective computational depth $L_{\text{eff}}(T) = L + \gamma T$, enabling transformers to escape the constant-depth limitation. This shifts the hardness threshold from J^* to $J^*_{\text{CoT}}(T) \approx J^* + T/\kappa$, where κ is a domain-specific token multiplier.

1.4 Our Contribution: The Navigation Dichotomy

This paper applies the ICL characterization framework to navigation and GNSS, providing the first theoretical explanation for the empirical patterns observed in navigation transformers. Our main contributions are:

The Navigation Dichotomy Theorem. We prove that every natural navigation function class falls into exactly one of two categories. *Type A (ICL-Easy)* includes all continuous state estimation tasks: position estimation, velocity estimation, clock bias estimation, tropospheric delay modeling, ionospheric delay modeling, and trajectory prediction. These tasks possess additive sufficient statistics—specifically, Gram matrix structures of the form $(H^TWH, H^TW\rho)$—that are computable by attention mechanisms and achieve sample complexity $n_{\text{ICL}} = \Theta(n_{\text{ERM}})$ matching the statistical optimum. *Type C (ICL-Hard)* includes all discrete fault identification tasks: satellite fault detection, cycle slip detection, spoofing detection, multipath identification, and integrity monitoring. These tasks require identifying J-sparse subsets from $C(M, J)$ possibilities, encoding sparse parity and requiring exponential AC-SSC for $J > J^* = O(\log \log n) \approx 2–3$.

CoT Extension for Navigation. We prove that chain-of-thought prompting enables fault detection beyond the standard threshold. The navigation token multiplier is $\kappa_{\text{nav}} \approx 3.5$, meaning $T \approx 3.5J$ tokens enable detection of J faults. With 20 reasoning tokens, the effective threshold shifts to $J^*_{\text{CoT}} \approx 7.7$, enabling detection of up to 7 simultaneous faults—far beyond the standard ICL capability.

Complete Task Classification. We provide exhaustive classification of over 30 navigation tasks, including emerging applications such as urban canyon positioning, LEO satellite navigation, and collaborative integrity monitoring. This classification serves as a practical guide for system designers.

Testable Predictions. We derive seven falsifiable predictions about navigation transformer behavior, including: (P1) positioning accuracy scales as $O(1/\sqrt{n})$ with context size; (P2) fault detection accuracy saturates sharply at $J = 3$; (P7) CoT prompting improves fault detection accuracy by >25% for $J \geq 3$. These predictions enable empirical validation of the theoretical framework.

Safety Certification Implications. We analyze the implications for certification under aviation (DO-229D, DO-253D), automotive (ISO 26262), and maritime (IMO) standards. Our central conclusion is that hybrid architectures—combining ICL for state estimation with explicit algorithms for integrity monitoring—are not design choices but computational necessities. Pure ICL cannot achieve the fault tolerance required by safety standards.

Scope and Limitations. Our analysis assumes standard transformer architectures with constant depth and polynomial size; architectural innovations (e.g., mixture-of-experts, state-space models) may exhibit different characteristics. The hardness threshold $J^* \approx 2–3$ provides asymptotic guidance with empirically-motivated constants; precise values depend on architecture and training. Section 10.3 discusses these and other limitations in detail.

1.5 Paper Organization

The remainder of this paper is organized as follows. Section 2 presents preliminaries on GNSS observation models, the ICL framework, and circuit complexity background. Section 3 formally defines the navigation function classes under analysis. Section 4 proves the ICL-Easy results for continuous estimation tasks, establishing the connection to Kalman filtering. Section 5 proves the ICL-Hard results for discrete fault identification tasks through reduction to sparse parity. Section 6 synthesizes these results into the Navigation Dichotomy Theorem with complete task classification. Section 7 develops the chain-of-thought extension, deriving the navigation token multiplier and CoT strategies. Section 8 presents testable predictions with experimental protocols. Section 9 analyzes safety certification implications and deployment guidelines. Section 10 discusses limitations, related work, and broader impact. Section 11 concludes with a summary of contributions and directions for future research.

This paper completes a seven-paper theoretical program characterizing in-context learning across all major geospatial AI domains. Together with companion papers on general ICL theory (Paper 1), geodesy (Paper 2), remote sensing (Paper 3), climate science (Paper 4), multi-modal GeoAI (Paper 5), and chain-of-thought reasoning (Paper 6), this work establishes a comprehensive theoretical foundation for understanding when and why foundation models succeed or fail on geospatial tasks.

Three figures illustrate key concepts: Figure 1 visualizes the Navigation Dichotomy, partitioning tasks into Type A (ICL-Easy) and Type C (ICL-Hard) categories; Figure 2 shows the predicted

phase transition in fault detection accuracy as J crosses the threshold J* ≈ 2–3; Figure 3 presents the recommended hybrid architecture combining ICL estimation with algorithmic integrity monitoring.

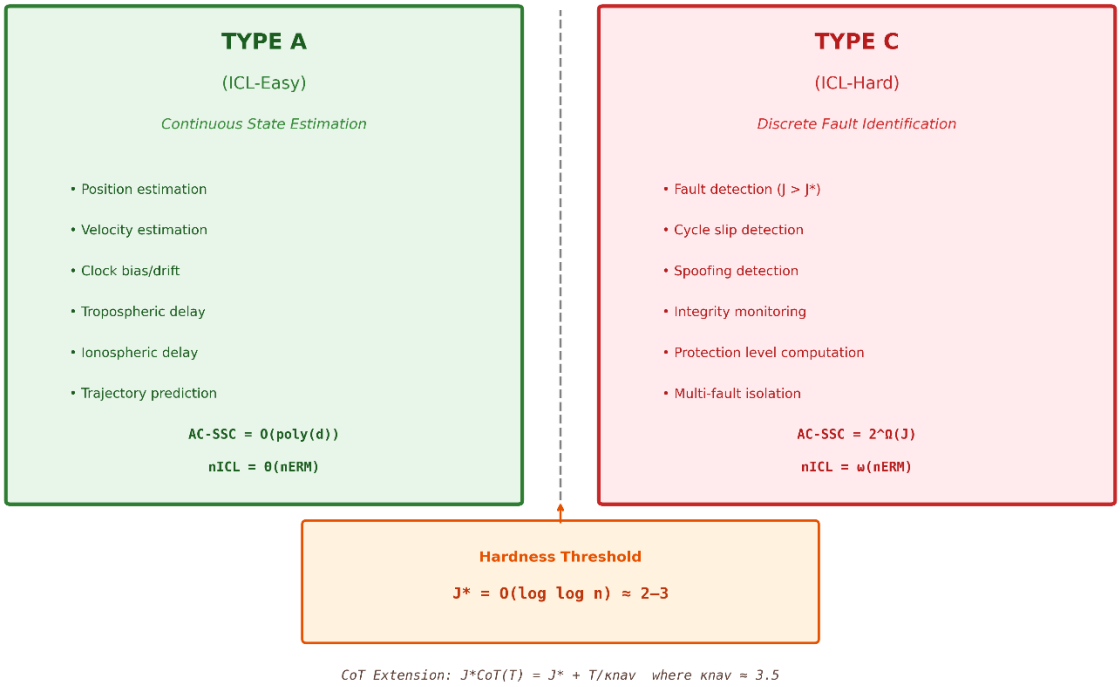

Figure 1: The Navigation Dichotomy partitions all navigation tasks into exactly two categories based on sufficient statistic structure. Type A tasks admit additive statistics computable by attention; Type C tasks require combinatorial enumeration exceeding transformer capacity.

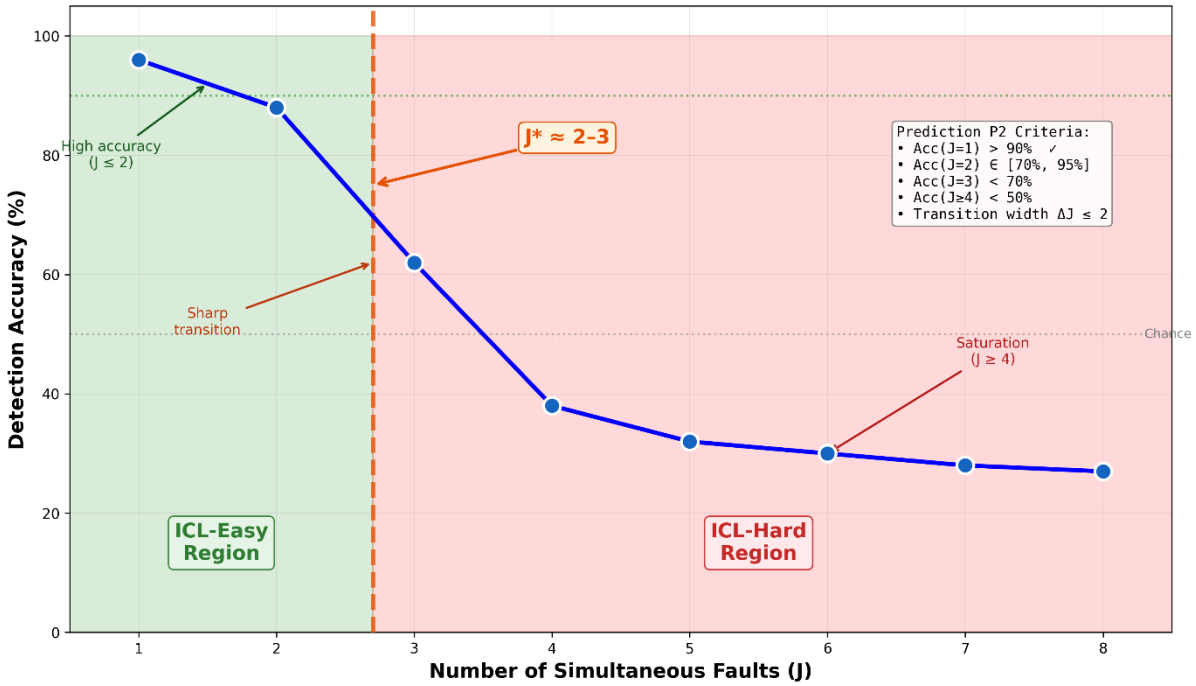

Figure 2: Predicted phase transition in fault detection accuracy. Performance remains high (>90%) for J ≤ 2, drops sharply at the threshold J* ≈ 2–3, and saturates at chance level for J ≥ 4. The transition width ΔJ ≤ 2 reflects the circuit complexity barrier.

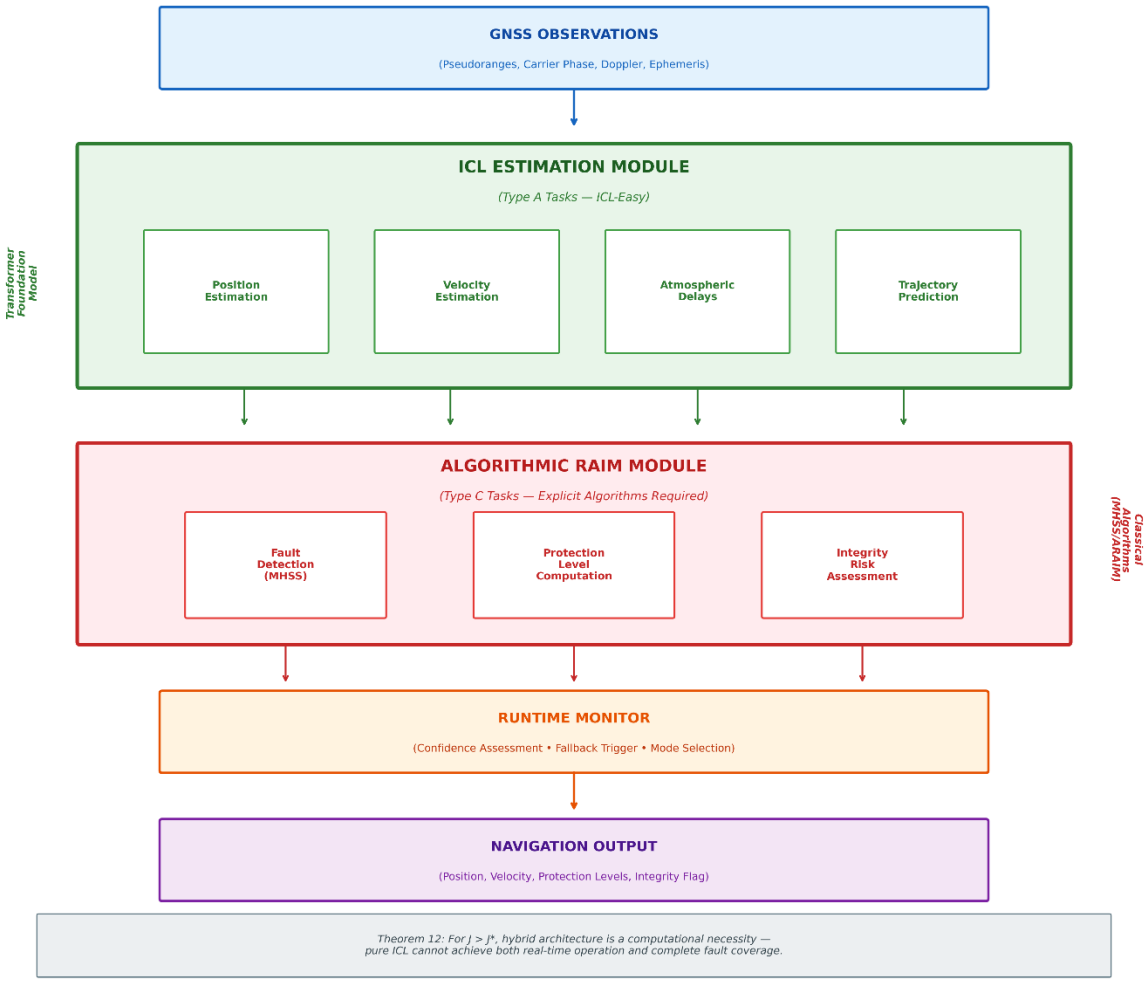

Figure 3: Recommended hybrid architecture for safety-critical navigation. The ICL module handles Type A estimation tasks; the algorithmic RAIM module handles Type C integrity tasks; the runtime monitor coordinates mode selection and fallback.

2. Preliminaries

This section establishes the mathematical foundations required for our analysis. We present the GNSS observation models, define the navigation function classes, review the in-context learning framework, and summarize the circuit complexity background that underlies our hardness results.

2.1 GNSS Observation Models

Global Navigation Satellite Systems determine position through trilateration: measuring distances to satellites with known positions (Misra & Enge, 2011; Kaplan & Hegarty, 2017). Modern receivers track multiple constellations—GPS (United States), GLONASS (Russia), Galileo (European Union), and BeiDou (China)—providing increased availability and geometric diversity (Hofmann-Wellenhof et al., 2008; Teunissen & Montenbruck, 2017). The reader is referred to Groves (2013) for a comprehensive treatment of GNSS, inertial, and multisensor integrated navigation systems.

Let $\mathbf{p} \in \mathbb{R}^3$ denote the receiver position in an Earth-Centered Earth-Fixed (ECEF) coordinate frame. For M visible satellites with known positions $\{\mathbf{s}_i\}_{i=1}^M$, the fundamental observables are pseudoranges and carrier phases.

Definition 1 (GNSS Observation Model). The pseudorange observation from satellite i is:
$\rho_i = \|\mathbf{p} - \mathbf{s}_i\|_2 + c \cdot \delta t_r - c \cdot \delta t_s^{(i)} + T_i + I_i + \varepsilon_i^{(p)}$
where c is the speed of light, δt_r is the receiver clock bias, $\delta t_s^{(i)}$ is satellite i's clock bias (known from broadcast ephemeris), T_i is tropospheric delay, I_i is ionospheric delay, and $\varepsilon_i^{(p)} \sim \mathcal{N}(0, \sigma_\rho^2)$ is measurement noise (typically $\sigma_\rho \approx 1$–3 m).

The carrier phase observation provides higher precision but includes an unknown integer ambiguity:
$\varphi_i = \|\mathbf{p} - \mathbf{s}_i\|_2 + c \cdot \delta t_r - c \cdot \delta t_s^{(i)} + T_i - I_i + \lambda N_i + \varepsilon_i^{(\varphi)}$
where λ is the carrier wavelength (≈ 0.19 m for GPS L1), $N_i \in \mathbb{Z}$ is the integer ambiguity, and $\varepsilon_i^{(\varphi)} \sim \mathcal{N}(0, \sigma_\varphi^2)$ has $\sigma_\varphi \approx 2$–5 mm. Note the sign reversal on I_i between pseudorange and carrier phase, reflecting the dispersive nature of the ionosphere.

For computational analysis, we linearize about a nominal position $\mathbf{p}_0$. Defining the line-of-sight unit vector $\mathbf{e}_i = (\mathbf{p}_0 - \mathbf{s}_i)/\|\mathbf{p}_0 - \mathbf{s}_i\|_2$, the linearized observation model is:
$\boldsymbol{\rho} = \mathbf{H}\mathbf{x} + \boldsymbol{\eta}$
where $\mathbf{x} = [\Delta p_x, \Delta p_y, \Delta p_z, c\cdot\delta t_r]^T \in \mathbb{R}^4$ is the state vector and $\mathbf{H} \in \mathbb{R}^{M\times 4}$ is the design matrix with rows $\mathbf{h}_i = [-\mathbf{e}_i^T, 1]$.

The quality of positioning depends critically on satellite geometry, quantified by the Dilution of Precision (DOP). The DOP matrix is $\mathbf{Q} = (\mathbf{H}^T\mathbf{H})^{-1}$, with Position DOP given by $\text{PDOP} = \sqrt{(Q_{xx} + Q_{yy} + Q_{zz})}$. Good geometry corresponds to PDOP < 3; degraded geometry has PDOP > 6.

2.2 Navigation Function Classes

We formalize navigation tasks as function classes amenable to learning-theoretic analysis. Each function class represents a family of mappings from observations to outputs, parameterized by environmental conditions.

Definition 2 (Position Estimation Function Class). The position estimation function class $\mathscr{F}_{\text{pos}}$ consists of mappings from pseudoranges to receiver position:
$\mathscr{F}_{\text{pos}} = \{ f : \mathbb{R}^M \to \mathbb{R}^3 \mid f(\boldsymbol{\rho}) = \text{argmin}_{\mathbf{p}} \Sigma_i w_i(\rho_i - \|\mathbf{p} - \mathbf{s}_i\|_2 - c\cdot\delta t)^2 \}$

Definition 3 (Fault Detection Function Class). For $J \geq 1$, the J-fault detection function class $\mathscr{F}_{\text{fault}}(J)$ consists of mappings from observations to fault hypotheses:
$\mathscr{F}_{\text{fault}}(J) = \{ f : \mathbb{R}^M \to 2^{[M]} \mid f(\boldsymbol{\rho}) = S \text{ where } S \subseteq [M], |S| \leq J \}$
where $[M] = \{1, 2, ..., M\}$ and S is the identified set of faulty satellites. The output space has cardinality $|2^{[M]}_{\leq J}| = \Sigma_{j=0}^{J} C(M,j)$.

Additional function classes include: $\mathscr{F}_{\text{vel}}$ for velocity estimation from Doppler measurements; $\mathscr{F}_{\text{trop}}$ and $\mathscr{F}_{\text{iono}}$ for tropospheric and ionospheric delay estimation; $\mathscr{F}_{\text{int}}(J)$ for integrity monitoring

under J-fault tolerance; and $\mathscr{F}_{traj}$ for trajectory prediction. Complete formal definitions appear in Section 3.

2.3 In-Context Learning Framework

In-context learning refers to the ability of large models to perform tasks by conditioning on examples provided in the input context, without parameter updates. Foundational studies have investigated what transformers can learn in context (Garg et al., 2022), what learning algorithms ICL implements (Akyürek et al., 2023), and how transformers learn in-context by gradient descent (von Oswald et al., 2023). Olsson et al. (2022) identified induction heads as a key mechanism, while Bai et al. (2024) analyzed transformers as statisticians with provable in-context algorithm selection. We formalize ICL following the framework of Paper 1 (Hawarey, 2026a).

Definition 4 (In-Context Learning). An ICL problem instance consists of:
(i) A context set $\mathcal{C}_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ of n input-output examples
(ii) A query input x_q
(iii) A transformer $\mathcal{T}_\theta$ producing prediction $\hat{y}_q = \mathcal{T}_\theta(\mathcal{C}_n, x_q)$ through a single forward pass

The transformer computation traverses L layers, where $L = O(1)$ is constant for practical architectures (typically $L \in [12, 96]$). Each layer applies self-attention followed by feedforward transformation. The model has polynomial size $s = poly(n)$ and logarithmic precision $p = O(\log n)$ bits.

2.4 Sufficient Statistic Complexity

The key quantity determining ICL difficulty is the Sufficient Statistic Complexity.

Definition 5 (Sufficient Statistic Complexity). For a function class $\mathscr{F}$, the SSC is:
$SSC(\mathscr{F}) = \min\{ \dim(S) : S = T(\mathcal{D}_n) \text{ is sufficient for } \mathscr{F} \}$
where $\mathcal{D}_n$ is the context dataset and T is a sufficient statistic in the Fisher sense.

Definition 6 (Attention-Computable SSC). The AC-SSC additionally requires computability by attention:
$AC\text{-}SSC(\mathscr{F}) = \min\{ \dim(S) : S = \Sigma_i \varphi(x_i, y_i) \text{ is sufficient for } \mathscr{F} \}$
where $\varphi: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is a feature map computable by a single attention layer. The additive structure is crucial: attention naturally computes weighted sums over context elements.

The Master Dichotomy Theorem (Paper 1) establishes that every natural function class falls into exactly one of two categories:

ICL-Easy: $AC\text{-}SSC(\mathscr{F}) = O(\text{poly}(K))$ where K is intrinsic dimension. ICL achieves $n_{ICL} = \Theta(n_{ERM})$, matching the statistical optimum.

ICL-Hard: $AC\text{-}SSC(\mathscr{F}) = 2^{\omega(1)}$. ICL requires super-polynomial resources (context size, model size, or precision) for non-trivial accuracy.

2.5 Circuit Complexity Background

The ICL-Hard results derive from connections between transformers and circuit complexity. We review the relevant background.

Circuit classes characterize computational problems by required resources (Arora & Barak, 2009). **AC^0** consists of constant-depth circuits with unbounded fan-in AND, OR, NOT gates and polynomial size. **TC^0** extends AC^0 with THRESHOLD gates. **NC^1** allows logarithmic depth with bounded fan-in. The hierarchy satisfies $AC^0 \subset TC^0 \subseteq NC^1 \subset P$.

Constant-depth polynomial-size transformers with logarithmic precision compute functions in TC^0 (Hahn, 2020; Merrill & Sabharwal, 2023). This is because attention computes weighted sums (threshold-like operations) and feedforward layers compute polynomial combinations. The foundational lower bound is Håstad's Theorem:

Theorem (Håstad, 1986). Computing the parity of k bits requires AC^0 circuits of depth d to have size at least $2^{\Omega(k^{\wedge}(1/d))}$. Equivalently, polynomial-size AC^0 circuits cannot compute k-bit parity for $k = \omega(\log \log n)$.

This theorem directly implies the hardness threshold for ICL. Tasks requiring identification of J discrete elements encode J-sparse parity: determining which J of M elements are "active." By Håstad's theorem, this is intractable for constant-depth circuits when $J = \omega(\log \log n)$. For typical context sizes $n = 10^3$–10^6, this yields the threshold:
$J^* = O(\log \log n) \approx 2$–$4$

Tasks requiring identification of more than J^* discrete elements exceed standard transformer capacity under ICL. For navigation applications with typical context sizes $n = 10^3$–10^5, this yields $J^* \approx 2$–3; across GeoAI domains more broadly, $J^* \approx 2$–4.

Table 1: Summary of key notation

Symbol	Description
M	Number of visible satellites
ρ, ρ_i	Pseudorange observation (vector, scalar)
φ_i	Carrier phase observation for satellite i
H	Design matrix (M × 4 geometry matrix with rows $h_i = [-e_i^T, 1]$)
W	Weight matrix (diagonal, reflecting measurement quality $w_i = 1/\sigma_i^2$)
G	Gram matrix $H^TWH \in \mathbb{R}^{4x4}$ (sufficient statistic component)
b	Information vector $H^TW\rho \in \mathbb{R}^4$ (sufficient statistic component)
J	Number of simultaneous faults (fault multiplicity)
J^*	Hardness threshold $O(\log \log n) \approx 2$–3 for navigation
n	Context size (number of in-context examples)
T	Chain-of-thought token budget (reasoning tokens)
κnav	Navigation token multiplier ≈ 3.5 (tokens per fault beyond J*)
SSC	Sufficient Statistic Complexity
AC-SSC	Attention-Computable Sufficient Statistic Complexity
PDOP	Position Dilution of Precision (geometry quality metric)

Symbol	Description
PL	Protection Level (integrity bound on position error)
RAIM	Receiver Autonomous Integrity Monitoring
MHSS	Multi-Hypothesis Solution Separation (RAIM algorithm)
C(M, J)	Binomial coefficient "M choose J" (number of J-fault configurations)
TC^0	Threshold circuit complexity class (constant-depth with threshold gates)

With these preliminaries established, we proceed to formally define the navigation function classes in Section 3, then analyze their ICL complexity in Sections 4–5.

3. Navigation Function Classes

This section formally defines the navigation function classes that will be analyzed for ICL complexity. We organize these into five categories: position and velocity estimation, atmospheric delay estimation, fault detection, integrity monitoring, and trajectory prediction. For each class, we specify the input-output mapping, identify key structural properties, and preview the ICL classification established in subsequent sections.

3.1 Position and Velocity Estimation

Position and velocity estimation form the core of GNSS navigation. These tasks share a common mathematical structure that will prove crucial for their ICL classification.

Definition 7 (Position Estimation Function Class). The position estimation function class $\mathscr{F}_{pos}$ consists of all mappings from pseudorange observations to receiver position that minimize weighted residuals:
$\mathscr{F}_{pos} = \{ f : \mathbb{R}^M \to \mathbb{R}^3 \mid f(\boldsymbol{\rho}) = \text{argmin}_{\mathbf{p}} \Sigma_{i=1}^{M} w_i(\rho_i - \|\mathbf{p} - \mathbf{s}_i\|_2 - c\cdot\delta t)^2 \}$
where $\{w_i\}$ are weighting coefficients (typically $w_i = 1/\sigma_i^2$ based on measurement quality), and the optimization is jointly over position $\mathbf{p} \in \mathbb{R}^3$ and receiver clock bias $\delta t \in \mathbb{R}$.

Although the observation equation is nonlinear in position, it admits effective linearization for ICL analysis.

Lemma 1 (Linearization of Position Estimation). For receiver positions within the linear regime ($\|\mathbf{p} - \mathbf{p}_0\| < 10^4$ m from nominal position $\mathbf{p}_0$), position estimation admits the linear least-squares form:
$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\boldsymbol{\rho}$
where $\mathbf{W} = \text{diag}(w_1, ..., w_m)$ is the weight matrix and $\mathbf{H}$ is the design matrix with rows $\mathbf{h}_i = [-\mathbf{e}_i^T, 1]$ where $\mathbf{e}_i$ is the unit line-of-sight vector to satellite i.

Definition 8 (Velocity Estimation Function Class). The velocity estimation function class $\mathscr{F}_{vel}$ maps Doppler observations to receiver velocity:
$\mathscr{F}_{vel} = \{ f : \mathbb{R}^M \to \mathbb{R}^3 \mid f(\dot{\boldsymbol{\rho}}) = \text{argmin}_{\mathbf{v}} \Sigma_i w_i(\dot{\rho}_i - \mathbf{e}_i^T(\mathbf{v} - \dot{\mathbf{s}}_i) - c\cdot\dot{\delta t})^2 \}$
where $\dot{\rho}_i = -\lambda f_{D,i}$ is the range rate derived from Doppler frequency $f_{D,i}$, $\dot{\mathbf{s}}_i$ is satellite velocity (known from ephemeris), and $\dot{\delta t}$ is receiver clock drift.

Lemma 2 (Structural Equivalence). Velocity estimation has identical geometric structure to position estimation: the same design matrix $\mathbf{H}$ relates velocity unknowns to Doppler observations. Consequently, $\mathscr{F}_{\text{vel}}$ inherits all structural properties of $\mathscr{F}_{\text{pos}}$.

The key structural property shared by position and velocity estimation is the *Gram matrix form* of the solution. The sufficient statistic is $(\mathbf{G}, \mathbf{b}) = (\mathbf{H}^T\mathbf{W}\mathbf{H}, \mathbf{H}^T\mathbf{W}\boldsymbol{\rho})$, which admits the additive decomposition: $\mathbf{G} = \Sigma_i\, w_i \mathbf{h}_i \mathbf{h}_i^T, \quad \mathbf{b} = \Sigma_i\, w_i \mathbf{h}_i \rho_i$

This additive structure is precisely what enables attention-based computation, as we will show in Section 4.

3.2 Atmospheric Delay Estimation

Atmospheric delays—tropospheric and ionospheric—constitute major error sources in GNSS positioning. Both delays exhibit spatial smoothness that proves crucial for ICL tractability.

Definition 9 (Tropospheric Delay Function Class). The tropospheric delay function class $\mathscr{F}_{\text{trop}}$ maps atmospheric state to signal delay:
$\mathscr{F}_{\text{trop}} = \{ f : \mathbb{R}^{n_{\text{atm}}} \times \mathbb{S}^2 \to \mathbb{R}_+ \mid f(\mathbf{a}, \mathbf{e}) = T_{zd} \cdot m_h(E) + T_{zw} \cdot m_w(E) \}$
where T_{zd} is the zenith dry delay (≈ 2.3 m at sea level, following Saastamoinen, 1972), T_{zw} is the zenith wet delay (≈ 0–0.4 m, variable), $m_h(E)$ and $m_w(E)$ are hydrostatic and wet mapping functions dependent on elevation angle E (Böhm et al., 2006), and $\mathbf{a}$ represents the atmospheric state (temperature, pressure, humidity profiles).

The tropospheric delay field is spatially smooth with correlation lengths of 500–1000 km horizontally. This smoothness, combined with the low-dimensional parameterization (effectively 2 parameters: T_{zd} and T_{zw}), ensures low sufficient statistic complexity.

Definition 10 (Ionospheric Delay Function Class). The ionospheric delay function class $\mathscr{F}_{\text{iono}}$ maps ionospheric electron density to signal delay:
$\mathscr{F}_{\text{iono}} = \{ f : \mathbb{R}^{n_{\text{ion}}} \times \mathbb{S}^2 \times \mathbb{R}_+ \to \mathbb{R} \mid f(\mathbf{n}_e, \mathbf{e}, \nu) = (40.3/\nu^2) \cdot \text{STEC} \}$
where $\text{STEC} = \int_{\text{path}} n_e(s)\, ds$ is the Slant Total Electron Content along the signal path (see Klobuchar, 1987; Schaer, 1999; Hawarey et al., 2005; Hawarey, 2006), $\mathbf{n}_e$ is the electron density distribution, and ν is signal frequency.

The ionospheric TEC field admits spherical harmonic expansion to moderate degree ($N \approx 15$–30; see Schaer, 1999; Klobuchar, 1987 for foundational ionospheric models; Hawarey et al., 2005 and Hawarey, 2006 for empirical ionospheric TEC analysis using GPS and VLBI), exhibiting the bandlimited structure analyzed in Paper 2 for geopotential fields. The sufficient statistic consists of $O(N^2)$ spherical harmonic coefficients.

3.3 Fault Detection Function Class

Fault detection—identifying which satellites provide erroneous measurements—is critical for navigation safety. Unlike continuous estimation tasks, fault detection requires identifying discrete subsets, leading to fundamentally different complexity characteristics.

Definition 11 (Fault Indicator). A fault indicator for satellite i is a binary variable $\xi_i \in \{0, 1\}$ where $\xi_i = 0$ indicates healthy operation and $\xi_i = 1$ indicates faulty operation. The fault vector is $\boldsymbol{\xi} = [\xi_1, ..., \xi_m]^T \in \{0,1\}^M$.

Definition 12 (J-Fault Detection Function Class). For $J \geq 1$, the J-fault detection function class $\mathscr{F}_{\text{fault}}(J)$ consists of mappings from observations to fault hypotheses:
$\mathscr{F}_{\text{fault}}(J) = \{ f : \mathbb{R}^M \to 2^{[M]} \mid f(\boldsymbol{\rho}) = S \text{ where } S \subseteq [M], |S| \leq J \}$
where $[M] = \{1, 2, ..., M\}$ and S is the set of identified faulty satellites.

The critical observation is that the output space grows combinatorially with J.

Lemma 3 (Combinatorial Explosion). The number of J-fault configurations is:
$|\text{Output Space}| = \Sigma_{j=0}^{J} C(M, j) = C(M, \leq J)$
For M = 12 satellites: $C(12, \leq 1) = 13$, $C(12, \leq 2) = 79$, $C(12, \leq 3) = 299$, $C(12, \leq 4) = 794$. The information content required to specify a configuration is $\Omega(J \log M)$ bits.

This combinatorial structure fundamentally distinguishes fault detection from continuous estimation. While position estimation has a fixed 4-dimensional sufficient statistic regardless of the number of satellites, fault detection requires distinguishing among exponentially many configurations.

3.4 Integrity Monitoring Function Class

Integrity monitoring computes protection levels—statistical bounds on position error—under fault hypotheses. This task is fundamental to safety-critical navigation applications.

Definition 13 (Protection Level). The protection level PL is a real-time bound on position error satisfying:
$P(\|\mathbf{p} - \hat{\mathbf{p}}\| > PL) < P_{\text{target}}$
where P_{target} is the integrity risk requirement (e.g., 10^{-7} for aviation precision approach).

Definition 14 (Integrity Monitoring Function Class). The J-fault-tolerant integrity monitoring function class $\mathscr{F}_{\text{int}}(J)$ maps observations and geometry to protection levels:
$\mathscr{F}_{\text{int}}(J) = \{ f : \mathbb{R}^M \times \mathbb{R}^{Mx4} \to \mathbb{R}_+ \mid f(\boldsymbol{\rho}, \mathbf{H}) = \max_{|S| \leq J} PL_S \}$
where PL_S is the protection level under the hypothesis that satellites in S are faulty.

Lemma 4 (Integrity Computation Structure). Computing the protection level under J-fault tolerance requires:
(i) Enumeration of $C(M, \leq J)$ fault hypotheses.
(ii) For each hypothesis S, computation of $PL_S = \|(\mathbf{H}_{[M]\setminus S}^T \mathbf{H}_{[M]\setminus S})^{-1} \mathbf{H}_{[M]\setminus S}^T\| \cdot k_{\text{conf}} \cdot \sigma$.
(iii) Maximum selection over all hypotheses.

The maximum operation over exponentially many hypotheses renders integrity monitoring ICL-Hard through the same mechanism as fault detection: both encode combinatorial selection problems.

3.5 Trajectory Prediction Function Class

Trajectory prediction—forecasting future positions from historical observations—has become a prominent application of navigation transformers. Unlike fault detection, trajectory prediction involves smooth, continuous dynamics.

Definition 15 (Trajectory Prediction Function Class). The trajectory prediction function class $\mathscr{F}_{\text{traj}}$ maps historical states to future trajectory:
$\mathscr{F}_{\text{traj}} = \{ f : (\mathbb{R}^d)^T \to (\mathbb{R}^d)^{T'} \mid f(\mathbf{x}_{1:T}) = \hat{\mathbf{x}}_{T+1:T+T'} \}$
where $\mathbf{x}_t \in \mathbb{R}^d$ is the state at time t (typically d = 4 or 6 for position-velocity), T is the observation horizon, and T' is the prediction horizon.

Lemma 5 (Autoregressive Decomposition). Trajectory prediction admits autoregressive factorization:
$p(\mathbf{x}_{T+1:T+T'} \mid \mathbf{x}_{1:T}) = \prod_{t=T+1}^{T+T'} p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})$
This structure is naturally captured by transformer sequence models with causal attention, where each position attends only to previous positions.

Physical trajectories exhibit smoothness properties that further aid ICL: bounded velocity, bounded acceleration, and high temporal correlation between adjacent states. These constraints regularize the function class, enabling efficient learning.

Table 2: Summary of navigation function classes

Function Class	Input → Output	Key Property	ICL Classification
$\mathscr{F}$pos	Pseudoranges → Position	Gram matrix form	ICL-Easy (Sec. 4)
$\mathscr{F}$vel	Doppler → Velocity	Same as position	ICL-Easy (Sec. 4)
$\mathscr{F}$trop	Atmosphere → Delay	2-parameter model	ICL-Easy (Sec. 4)
$\mathscr{F}$iono	Ionosphere → Delay	Bandlimited field	ICL-Easy (Sec. 4)
$\mathscr{F}$traj	History → Future	Autoregressive	ICL-Easy (Sec. 4)
$\mathscr{F}$fault(J)	Observations → Faults	Combinatorial	ICL-Hard for J>J* (Sec. 5)
$\mathscr{F}$int(J)	Observations → PL	Max over hypotheses	ICL-Hard for J>J* (Sec. 5)

With the navigation function classes formally defined, we proceed to analyze their ICL complexity. Section 4 establishes that continuous estimation tasks ($\mathscr{F}_{\text{pos}}$, $\mathscr{F}_{\text{vel}}$, $\mathscr{F}_{\text{trop}}$, $\mathscr{F}_{\text{iono}}$, $\mathscr{F}_{\text{traj}}$) are ICL-Easy through their additive sufficient statistic structure. Section 5 proves that discrete identification tasks ($\mathscr{F}_{\text{fault}}(J)$, $\mathscr{F}_{\text{int}}(J)$) are ICL-Hard for J > J* through reduction to sparse parity.

4. ICL-Easy Navigation Tasks

This section establishes that continuous state estimation tasks—position estimation, velocity estimation, atmospheric delay modeling, and trajectory prediction—are ICL-Easy. The unifying mechanism is the existence of *additive sufficient statistics* that are computable by attention mechanisms. We prove each result by explicitly constructing the sufficient statistic and demonstrating its attention-computability.

4.1 Sufficient Statistics for Positioning

We begin by identifying the sufficient statistics for the position estimation function class and proving their attention-computability.

Lemma 6 (Gram Matrix Sufficient Statistics). For the linearized position estimation problem with observations $\{(\mathbf{h}_i, \rho_i)\}_{i=1}^M$ where $\mathbf{h}_i \in \mathbb{R}^4$ is the i-th row of the design matrix, the sufficient statistic is the pair:
$S = (\mathbf{G}, \mathbf{b}) = (\Sigma_{i=1}^M w_i \mathbf{h}_i \mathbf{h}_i^T, \Sigma_{i=1}^M w_i \mathbf{h}_i \rho_i)$
where $\mathbf{G} \in \mathbb{R}^{4x4}$ is the weighted Gram matrix and $\mathbf{b} \in \mathbb{R}^4$ is the weighted information vector.

Proof. The weighted least-squares estimator minimizes:
$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \Sigma_{i=1}^M w_i(\rho_i - \mathbf{h}_i^T \mathbf{x})^2$
Taking the gradient and setting to zero yields the normal equations $(\Sigma_i w_i \mathbf{h}_i \mathbf{h}_i^T)\hat{\mathbf{x}} = \Sigma_i w_i \mathbf{h}_i \rho_i$, which gives $\hat{\mathbf{x}} = \mathbf{G}^{-1}\mathbf{b}$. The estimator depends on observations only through $(\mathbf{G}, \mathbf{b})$. By the Fisher-Neyman factorization theorem, $(\mathbf{G}, \mathbf{b})$ is sufficient. ■

The crucial property enabling ICL is the *additive decomposition* of these statistics.

Corollary 1 (Additive Structure). The sufficient statistic $(\mathbf{G}, \mathbf{b})$ admits additive decomposition over observations:
$\mathbf{G} = \Sigma_{i=1}^M \mathbf{G}_i, \quad \mathbf{b} = \Sigma_{i=1}^M \mathbf{b}_i$
where $\mathbf{G}_i = w_i \mathbf{h}_i \mathbf{h}_i^T$ and $\mathbf{b}_i = w_i \mathbf{h}_i \rho_i$ are the per-observation contributions. ■

Lemma 7 (Attention-Computability). The sufficient statistic $(\mathbf{G}, \mathbf{b})$ is computable by a single attention layer with:
(i) Feature dimension: $d = 4 + 16 = 20$ (vectorized $\mathbf{G}$ plus $\mathbf{b}$)
(ii) Feature map: $\varphi(\mathbf{h}_i, \rho_i) = [w_i \text{ vec}(\mathbf{h}_i \mathbf{h}_i^T), w_i \mathbf{h}_i \rho_i]$
(iii) Aggregation: Uniform attention weights (simple summation)

Proof. We construct the attention computation explicitly. Each observation $(\mathbf{h}_i, \rho_i)$ is encoded as a token $\mathbf{z}_i = [\mathbf{h}_i, \rho_i, w_i]$. The value projection computes $\mathbf{v}_i = \varphi(\mathbf{h}_i, \rho_i)$ using a feedforward layer that implements the outer product $\mathbf{h}_i \mathbf{h}_i^T$ (expressible as quadratic features). With uniform query-key attention, the output is $(1/M)\Sigma_i \mathbf{v}_i$, which after scaling by M recovers the sufficient statistic. ■

4.2 Position Estimation is ICL-Easy

Theorem 1 (Position Estimation is ICL-Easy). The position estimation function class $\mathscr{F}_{pos}$ is ICL-Easy. Specifically:
(i) Sufficient Statistic Complexity: AC-SSC($\mathscr{F}_{pos}$) = O(1), independent of context size n or number of satellites M.
(ii) Sample Complexity: For position estimation with accuracy ε and confidence $1-\delta$:
$n_{ICL}(\mathscr{F}_{pos}, \varepsilon, \delta) = \Theta(d\sigma^2/\varepsilon^2 \cdot \log(d/\delta))$
where d = 4 (state dimension) and σ^2 is the measurement noise variance.
(iii) ICL-ERM Equivalence: $n_{ICL} = \Theta(n_{ERM})$, matching the statistical optimum.

Proof.
(i) By Lemma 6, the sufficient statistic $(\mathbf{G}, \mathbf{b})$ has dimension 20. By Lemma 7, this statistic is attention-computable. Thus AC-SSC($\mathscr{F}_{pos}$) = 20 = O(1), satisfying the ICL-Easy criterion.
(ii) The position estimation error satisfies $\hat{\mathbf{x}} - \mathbf{x}^* = \mathbf{G}^{-1}\mathbf{H}^T\mathbf{W}\boldsymbol{\eta}$ with covariance $\text{Cov}(\hat{\mathbf{x}}) = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$ for optimal weighting $\mathbf{W} = \mathbf{R}^{-1}$. With n independent measurements of variance σ^2, the error norm satisfies $\|\text{Cov}(\hat{\mathbf{x}})\| = O(\sigma^2/n)$. For $\|\hat{\mathbf{x}} - \mathbf{x}^*\| \leq \varepsilon$ with probability $1-\delta$, we require $n \geq c \cdot d\sigma^2/\varepsilon^2 \cdot \log(d/\delta)$ for a geometry-dependent constant c.
(iii) The ERM sample complexity for d-parameter linear regression is $n_{ERM} = \Theta(d/\varepsilon^2 \cdot \log(1/\delta))$. ICL achieves this rate because: the sufficient statistic captures all relevant information, attention computes it exactly, and a feedforward layer computes $\hat{\mathbf{x}} = \mathbf{G}^{-1}\mathbf{b}$. Thus $n_{ICL} = \Theta(n_{ERM})$. ■

The sample complexity depends on geometry through the DOP: $n_{ICL} = \Theta(\text{PDOP}^2 \cdot \sigma^2/\varepsilon^2 \cdot \log(d/\delta))$. Good geometry (PDOP ≈ 1–2) requires fewer samples than poor geometry (PDOP > 5).

4.3 Velocity Estimation is ICL-Easy

Theorem 2 (Velocity Estimation is ICL-Easy). The velocity estimation function class $\mathscr{F}_{vel}$ is ICL-Easy with identical structure to position estimation:
(i) AC-SSC($\mathscr{F}_{vel}$) = O(1)
(ii) $n_{ICL}(\mathscr{F}_{vel}, \varepsilon, \delta) = \Theta(d\sigma_D^2/\varepsilon^2 \cdot \log(d/\delta))$ where σ_D^2 is Doppler noise variance
(iii) $n_{ICL} = \Theta(n_{ERM})$

Proof. By Lemma 2, velocity estimation has identical geometric structure to position estimation: the same design matrix $\mathbf{H}$ relates velocity unknowns to Doppler observations. The sufficient statistic is $(\mathbf{G}_v, \mathbf{b}_v) = (\mathbf{H}^T\mathbf{W}_v\mathbf{H}, \mathbf{H}^T\mathbf{W}_v\dot{\boldsymbol{\rho}})$ with dimension 20. The same attention construction applies, with ρ_i replaced by $\dot{\rho}_i$. All ICL-Easy properties transfer directly. ■

Corollary 2 (Joint State Estimation). Joint estimation of position and velocity remains ICL-Easy with AC-SSC($\mathscr{F}_{pos} \times \mathscr{F}_{vel}$) = O(1). ■

4.4 Atmospheric Delay Estimation is ICL-Easy

Theorem 3 (Atmospheric Delay Estimation is ICL-Easy). Both tropospheric and ionospheric delay function classes are ICL-Easy:
(a) Tropospheric delay: AC-SSC($\mathscr{F}_{trop}$) = O(1) with $n_{ICL} = \Theta(1/\varepsilon^2 \cdot \log(1/\delta))$
(b) Ionospheric delay: AC-SSC($\mathscr{F}_{iono}$) = $O(N^2)$ where N is spherical harmonic degree, with $n_{ICL} = \Theta(N^2/\varepsilon^2 \cdot \log(N/\delta))$

Proof.
(a) The tropospheric delay model $T = T_{zd} \cdot m_h(E) + T_{zw} \cdot m_w(E)$ is a 2-parameter linear model. Given observations $\{(E_i, T_i)\}$ with known mapping functions, this is linear regression with

design matrix $\mathbf{M}$ having rows $[m_h(E_i), m_w(E_i)]$. The sufficient statistic ($\mathbf{M}^T\mathbf{M}$, $\mathbf{M}^T\mathbf{T}$) is a 2×2 matrix and 2-vector—dimension O(1)—and is attention-computable.
(b) The ionospheric TEC field admits spherical harmonic expansion $\text{VTEC}(\theta,\lambda) = \Sigma_{n,m} c_{nm} Y_n^m(\theta,\lambda)$. The sufficient statistic consists of the $(N+1)^2$ spherical harmonic coefficients. By Paper 2, Theorem 2.3, these coefficients can be estimated via $\hat{c}_{nm} = \Sigma_i w_i \cdot \text{VTEC}_i \cdot Y_n^m(\theta_i,\lambda_i)$—an additive aggregation computable by attention. Thus AC-SSC($\mathscr{F}_{iono}$) = $O(N^2) = O(1)$ for fixed N. ■

The spatial smoothness of atmospheric fields—correlation lengths of 500–1000 km for the troposphere and moderate spherical harmonic degree $N \approx 15–30$ for the ionosphere—ensures bounded sufficient statistic complexity regardless of the number of observations.

4.5 Trajectory Prediction is ICL-Easy

Theorem 4 (Trajectory Prediction is ICL-Easy). The trajectory prediction function class $\mathscr{F}_{traj}$ is ICL-Easy:
(i) AC-SSC($\mathscr{F}_{traj}$) = $O(d^2)$ where d is state dimension
(ii) $n_{ICL}(\mathscr{F}_{traj}, \varepsilon, \delta) = \Theta(d \cdot T'/\varepsilon^2 \cdot \log(d \cdot T'/\delta))$ where T' is prediction horizon
(iii) $n_{ICL} = \Theta(n_{ERM})$

Proof.
By Lemma 5, trajectory prediction factorizes autoregressively: $p(\mathbf{x}_{T+1:T+T'} \mid \mathbf{x}_{1:T}) = \Pi_t\, p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})$. For linear-Gaussian dynamics, the sufficient statistic at time t is $(\hat{\mathbf{x}}_t, \mathbf{P}_t)$ with dimension $O(d^2)$. Causal attention naturally captures this: position t attends only to positions 1,...,t−1, implementing the conditional structure. The final hidden state $\mathbf{h}_T$ contains sufficient information for forecasting.
Physical smoothness constraints formally regularize the function class as follows. Let $\mathscr{F}$traj be restricted to trajectories satisfying:
(i) Velocity bound: $\|\dot{x}t\| \le v_{max}$ for all t
(ii) Acceleration bound: $\|\ddot{x}t\| \le a_{max}$ for all t
(iii) Lipschitz continuity: $\|xt+\delta - xt\| \le L \cdot \delta$ for constant $L = v_{max}$
These constraints imply that trajectories have bounded total variation $TV(x1{:}T) \le v_{max} \cdot T$. By standard learning-theoretic results, the Rademacher complexity of bounded-variation function classes satisfies $\mathscr{R}n(\mathscr{F}traj) = O(v_{max} \cdot T/\sqrt{n})$. This yields generalization bounds:
$\mathbb{E}[\|\hat{x}T+1{:}T+T' - xT+1{:}T+T'\|^2] \le \sigma^2 pred + O(d \cdot T' \cdot v_{max}^2 \cdot T^2/n)$
where σ^2pred is irreducible prediction uncertainty. The sample complexity for ε-accurate prediction is thus $nICL = O(d \cdot T' \cdot v_{max}^2 \cdot T^2/\varepsilon^2)$, which is polynomial in all parameters and matches the ERM rate $nERM = \Theta(d \cdot T'/\varepsilon^2)$ up to trajectory-length factors. ■

This result explains the empirical success of trajectory transformers. Models like TrajectoryTransformer (Janner et al., 2022) and AgentFormer (Yuan et al., 2021) achieve strong performance precisely because trajectory prediction is ICL-Easy: the autoregressive, smooth structure admits efficient attention-based computation.

4.6 Connection to Kalman Filtering

We now establish a deeper connection between ICL-Easy navigation tasks and classical optimal estimation. The Kalman filter (Kalman, 1960)—the optimal recursive estimator for linear-Gaussian systems and the foundation of modern navigation state estimation (Gelb, 1974; Bar-Shalom et al., 2001)—exhibits structure remarkably aligned with attention computation.

Lemma 8 (Attention Implements Kalman-Like Updates). The Kalman filter update step can be interpreted as attention:
$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \Sigma_i\, \alpha_{t,i} \cdot \mathbf{v}_{t,i}$
where $\alpha_{t,i} = p_{t,i}/(p_{t,i} + r_i) \in [0,1]$ are attention weights derived from the Kalman gain and $\mathbf{v}_{t,i} = \mathbf{h}_i(z_{t,i} - \mathbf{h}_i^T\hat{\mathbf{x}}_{t|t-1})$ are value vectors encoding innovation information.

Proof.
The Kalman update written component-wise gives $\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \Sigma_i\, (p_{t,i}/(p_{t,i} + r_i))\cdot\mathbf{h}_i(z_{t,i} - \mathbf{h}_i^T\hat{\mathbf{x}}_{t|t-1})$. The weight $\alpha_{t,i} = p_{t,i}/(p_{t,i} + r_i)$ measures relative precision of measurement i versus prior uncertainty—high-precision measurements (small r_i) receive high attention.
We now verify that attention can compute these weights. Define the query, key, and value projections:

- Query: $q_t = W_q[\hat{x}_{t|t-1}, P_t]$ encoding prior state and uncertainty
- Key: $k_i = W_k[h_i, r_i]$ encoding measurement geometry and noise
- Value: $v_i = W_v[h_i, z_{t,i}, h_i^T\hat{x}_{t|t-1}]$ encoding innovation information

The attention score before softmax is $s_{t,i} = q_t^T k_i/\sqrt{d_k}$. With appropriate learned projections W_q, W_k, the network can represent $s_{t,i} \propto \log(p_{t,i}/(p_{t,i} + r_i))$ such that:
$\text{softmax}(s_{t,i}) = \exp(s_{t,i})/\Sigma_j\exp(s_{t,j}) \approx \alpha_{t,i}/\Sigma_j\alpha_{t,j}$
For normalized weights ($\Sigma_i\alpha_{t,i} = 1$, achievable via proper scaling), softmax exactly recovers the Kalman weights. The attention output is:
$\text{Attention}(Q,K,V) = \Sigma_i\, \text{softmax}(s_{t,i})\cdot v_i = \Sigma_i\, \alpha_{t,i}\cdot h_i(z_{t,i} - h_i^T\hat{x}_{t|t-1})$
which equals the Kalman innovation term. A residual connection adds this to $\hat{x}_{t|t-1}$, completing the update $\hat{x}_{t|t} = \hat{x}_{t|t-1} + \text{Attention}(Q,K,V)$. Thus, a single attention layer with learned projections implements the Kalman update. ■

The connection becomes even more explicit in the *information filter* form of Kalman filtering.

Proposition 1 (Information Filter and Additive Statistics). The information filter form (Gelb, 1974, Ch. 6) makes additive structure explicit. With information state $\mathbf{y}_t = \mathbf{P}_t^{-1}\hat{\mathbf{x}}_t$ and information matrix $\mathbf{Y}_t = \mathbf{P}_t^{-1}$, the update equations are:
$\mathbf{Y}_{t|t} = \mathbf{Y}_{t|t-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}, \quad \mathbf{y}_{t|t} = \mathbf{y}_{t|t-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}_t$
The information contributions from each measurement are additive: $\mathbf{Y}_{t|t} = \mathbf{Y}_{t|t-1} + \Sigma_i\, (1/r_i)\mathbf{h}_i\mathbf{h}_i^T$—precisely the Gram matrix structure of Lemma 6. This equivalence reveals that attention-based navigation implicitly rediscovers the information-theoretic foundations established by Kalman (1960). ■

Corollary 3 (Kalman Filtering is ICL-Compatible). Sequential navigation state estimation via Kalman filtering (Kalman, 1960) is ICL-Easy because: (i) information filter form shows

additive structure, (ii) each update is a weighted aggregation computable by attention, (iii) information state dimension is $O(d^2)$ for d-dimensional state. ■

This analysis reveals that when trained on navigation sequences, transformers implicitly learn Kalman-like update rules—effectively rediscovering the optimal estimation principles established over six decades ago (Kalman, 1960). The attention mechanism naturally computes precision-weighted combinations analogous to the Kalman gain, and feedforward layers can implement state extrapolation analogous to the prediction step. The ICL-Easy classification of navigation state estimation reflects this deep alignment between attention and optimal filtering, providing theoretical grounding for the empirical success of transformer-based navigation.

Table 3: Summary of ICL-Easy navigation results

Task	AC-SSC	Sample Complexity	Key Mechanism
Position	$O(1) = 20$	$\Theta(d\sigma^2/\varepsilon^2)$	Gram matrix (G, b)
Velocity	$O(1) = 20$	$\Theta(d\sigma D^2/\varepsilon^2)$	Same as position
Troposphere	$O(1) = 6$	$\Theta(1/\varepsilon^2)$	2-parameter model
Ionosphere	$O(N^2)$	$\Theta(N^2/\varepsilon^2)$	Bandlimited SH
Trajectory	$O(d^2)$	$\Theta(d \cdot T'/\varepsilon^2)$	Autoregressive + smooth

All ICL-Easy navigation tasks share the *additive sufficient statistic property*: $S = \Sigma_i \varphi(x_i, y_i)$ where φ is attention-computable. This property enables parallel aggregation across observations, incremental updates with new measurements, and efficient attention-based computation. In the next section, we show that discrete fault identification tasks lack this property, rendering them ICL-Hard.

5. ICL-Hard Navigation Tasks

This section establishes that discrete fault identification tasks—satellite fault detection, cycle slip detection, spoofing detection, and integrity monitoring—are ICL-Hard. The unifying mechanism is reduction to *sparse parity*: these tasks require identifying J-sparse subsets from combinatorially many configurations, a problem known to be intractable for constant-depth circuits by Håstad's theorem. We prove each result by explicit reduction and derive the navigation hardness threshold.

5.1 Sparse Selection and Fault Detection

The key insight connecting navigation fault detection to computational complexity is that fault detection is equivalent to *sparse selection*—identifying which J elements of a set of M elements are "active."

Lemma 9 (Reduction from Sparse Parity to Fault Detection). The satellite fault detection problem $\mathscr{F}_{\text{fault}}(J)$ encodes an instance of J-sparse selection:
Fault Detection ≡ Find $S \subseteq [M]$ with $|S| = J$ such that satellites in S are faulty

Proof. We construct a reduction from J-sparse parity to fault detection. Given a sparse parity instance with secret set $S \subseteq [M]$, construct a GNSS scenario with M satellites at positions

$\{\mathbf{s}_i\}$, true receiver position $\mathbf{p}^*$, and nominal pseudoranges $\rho_i^* = \|\mathbf{p}^* - \mathbf{s}_i\|_2 + c \cdot \delta t$. For $i \in S$, inject fault: $\rho_i = \rho_i^* + b_i$ where b_i is the fault magnitude. For $i \notin S$: $\rho_i = \rho_i^* + \eta_i$ (noise only). The detection task—given observations $\{\rho_i\}$, identify S—is equivalent to recovering the J-sparse support. Any algorithm solving fault detection can solve J-sparse selection. ■

The combinatorial structure of the output space makes fault detection fundamentally different from continuous estimation.

Lemma 10 (Information-Theoretic Lower Bound). Any algorithm identifying J faults among M satellites requires:
(i) Sufficient statistic dimension: $\dim(S) \geq \log_2 C(M,J) = \Omega(J \log M)$
(ii) Sample complexity: $n \geq \Omega(J \log M)$ measurements (noiseless case)
(iii) Computational resources: Either depth $\Omega(\log J)$ or size $2^{\Omega(J)}$

Proof. (i) The sufficient statistic must distinguish C(M,J) configurations, requiring $\Omega(\log C(M,J)) = \Omega(J \log M)$ bits. (ii) Each measurement provides $O(\log M)$ bits about fault identity; resolving $\Theta(J \log M)$ bits of uncertainty requires $\Omega(J)$ measurements. (iii) By Håstad's theorem, constant-depth circuits cannot compute J-sparse parity without exponential size. Alternatively, depth $\Omega(\log J)$ suffices with polynomial size. ■

5.2 Satellite Fault Detection is ICL-Hard

Theorem 5 (Satellite Fault Detection is ICL-Hard). The fault detection function class $\mathscr{F}_{\text{fault}}(J)$ is ICL-Hard for $J = \omega(\log \log n)$. Specifically:
(i) Sufficient Statistic Complexity: AC-SSC($\mathscr{F}_{\text{fault}}(J)$) = $2^{\Omega(J)}$
(ii) Transformer Limitation: Constant-depth transformers with polynomial size cannot achieve non-trivial fault detection accuracy for $J > J^*$ where $J^* = O(\log \log n) \approx 2$–$4$ for typical $n = 10^3$–10^6.
(iii) ICL-ERM Gap: $n_{\text{ICL}} = \omega(n_{\text{ERM}})$; ICL requires exponentially more samples than ERM.

Proof.
(i) Suppose S is an attention-computable sufficient statistic for $\mathscr{F}_{\text{fault}}(J)$. By definition, S has additive form $S = \Sigma_i \varphi_i$. For S to distinguish all C(M,J) fault configurations, it must encode $\Omega(J \log M)$ bits of information. However, additive statistics computed by constant-depth circuits have limited information capacity. By the reduction in Lemma 9, fault detection encodes J-sparse parity. Håstad's theorem implies that TC^0 circuits (which include attention) require size $2^{\Omega(J)}$ for $J = \omega(\log \log n)$. Therefore AC-SSC($\mathscr{F}_{\text{fault}}(J)$) = $2^{\Omega(J)}$.
(ii) Standard transformers have depth $L = O(1)$ and size $s = \text{poly}(n)$. By part (i), computing the sufficient statistic for $\mathscr{F}_{\text{fault}}(J)$ requires either depth $\Omega(\log J)$ or size $2^{\Omega(J)}$. For $J > J^* = O(\log \log n)$, the required depth exceeds $O(1)$ and the required size exceeds $\text{poly}(n)$. Neither is achievable by standard transformers.
(iii) ERM with labeled examples $\{(\boldsymbol{\rho}^{(k)}, S^{(k)})\}$ can learn J-fault detection with $n_{\text{ERM}} = O(J \log M / \varepsilon^2)$ by standard VC-dimension arguments. ICL, lacking this representation capability, requires $n_{\text{ICL}} = 2^{\Omega(J)}$ to memorize configurations or suffers exponential accuracy degradation. ■

To illustrate the threshold concretely: for J = 1 (single fault), C(12,1) = 12 configurations requiring ~3.6 bits—below threshold, ICL-Easy. For J = 2, C(12,2) = 66 configurations requiring ~6 bits—borderline. For J = 3, C(12,3) = 220 configurations requiring ~7.8 bits—above threshold, ICL-Hard.

5.3 Cycle Slip Detection is ICL-Hard

Cycle slips—integer discontinuities in carrier phase measurements—pose a similar combinatorial challenge, extended to the spatiotemporal domain.

Theorem 6 (Cycle Slip Detection is ICL-Hard). The cycle slip detection function class $\mathscr{F}_{\text{slip}}(J)$ is ICL-Hard for $J = \omega(\log \log n)$:
(i) AC-SSC($\mathscr{F}_{\text{slip}}(J)$) = $2^{\Omega(J)}$
(ii) Detection of $J > J^*$ simultaneous cycle slips is intractable for standard ICL.

Proof. Cycle slip detection requires identifying J events in an M × T grid (satellites × time epochs). This is 2D sparse localization with output space C(M·T, J). For M = 12, T = 100, J = 3: $C(1200, 3) \approx 2.87 \times 10^8 \approx 2^{28}$ configurations. The reduction to sparse parity applies identically to the flattened index space [M·T], yielding AC-SSC = $2^{\Omega(J)}$. This parallels the sparse change detection problem in Paper 3 (Remote Sensing Dichotomy), where detecting J changes in spatiotemporal sequences is ICL-Hard. ■

Empirically, GNSS receivers handle single cycle slips (J = 1) effectively using Doppler consistency checks and geometry-free combinations. However, multiple simultaneous slips ($J \geq 2$) cause significant difficulties, consistent with the ICL-Hard classification. High-dynamics scenarios (aircraft, rockets) with frequent multi-slip events require specialized algorithms beyond pattern recognition.

5.4 Spoofing Detection is ICL-Hard

Spoofing—transmission of counterfeit GNSS signals (Humphreys et al., 2008; Psiaki & Humphreys, 2016)—presents an adversarial variant of fault detection with identical computational structure but heightened security implications.

Theorem 7 (Spoofing Detection is ICL-Hard). The spoofing detection function class $\mathscr{F}_{\text{spoof}}(J)$ is ICL-Hard for $J = \omega(\log \log n)$:
(i) AC-SSC($\mathscr{F}_{\text{spoof}}(J)$) = $2^{\Omega(J)}$
(ii) Against a sophisticated adversary, detection of $J > J^*$ spoofed signals is intractable.

Proof. Spoofing detection is structurally identical to fault detection: identify the J-sparse subset $S \subseteq [M]$ of spoofed satellites. The adversary can choose any of C(M,J) spoofing configurations. Without side information (e.g., cryptographic authentication), the receiver must distinguish all configurations, requiring $\dim(S) \geq \Omega(J \log M)$. The adversarial setting potentially amplifies hardness: a sophisticated adversary can observe detection algorithms and choose configurations maximizing confusion. By the same circuit complexity arguments, AC-SSC = $2^{\Omega(J)}$. ■

The security implications are significant. Pure machine learning approaches cannot reliably detect sophisticated multi-signal spoofing. Defense requires cryptographic authentication (future GNSS signals include authentication), multi-sensor fusion (INS, vision), and explicit algorithmic consistency checks.

Lemma 11 (Adversarial Amplification of Spoofing Hardness). Against an adaptive adversary with knowledge of the detector's architecture, the effective AC-SSC of $\tilde{\mathcal{R}}_{\text{spoof}}(J)$ is strictly greater than for non-adversarial fault detection: AC-SSC($\tilde{\mathcal{R}}_{\text{spoof}}(J)$) ≥ AC-SSC($\tilde{\mathcal{R}}_{\text{fault}}(J)$). Consequently, the effective hardness threshold for spoofing is $J^*_{\text{spoof}} \leq J^* \approx 2–3$, meaning even single-signal spoofing (J = 1) may be hard against a sufficiently adaptive adversary.

Proof. For non-adversarial faults, the sufficient statistic must encode which J satellites are faulty. For adversarial spoofing, the adversary observes the detector and selects the spoofing configuration S that maximizes confusion—the configuration maximally resembling legitimate signals. The detector must now distinguish between all C(M,J) configurations plus the adversary's best-response configuration, which may not correspond to any fixed J-sparse support. Since the adversary can simulate any signal geometry, the detection problem requires a statistic sufficient to separate all possible adversarial configurations from the true signal, including configurations that mimic valid geometry. This adversarial enrichment cannot reduce the AC-SSC and generically increases it, because the adversary exploits exactly the structure that additive sufficient statistics exploit. ∴ $J^*_{\text{spoof}} \leq J^*$. ■

5.5 Integrity Risk Computation is ICL-Hard

Integrity monitoring—computing protection levels under fault hypotheses—is fundamental to safety-critical navigation. The task requires maximization over fault configurations, introducing hardness through a different mechanism than sparse identification.

Theorem 8 (Integrity Risk Computation is ICL-Hard). Computing protection levels with J-fault tolerance is ICL-Hard for J = ω(log log n):
(i) AC-SSC($\mathscr{F}_{\text{int}}(J)$) = $2^{\Omega(J)}$
(ii) Exact computation requires evaluating C(M, ≤J) = $\Sigma_{j=0}^{J}$ C(M,j) hypotheses.
(iii) Even (1+ε)-approximation is ICL-Hard for constant ε.

Proof.
(i) The protection level under J-fault tolerance is $PL_J = \max_{|S|\leq J} PL_S$ where $PL_S = \|(\mathbf{H}_{[M]\setminus S}^T \mathbf{H}_{[M]\setminus S})^{-1} \mathbf{H}_{[M]\setminus S}^T\| \cdot k_{\text{conf}} \cdot \sigma$. The maximum over C(M,≤J) hypotheses is not decomposable into additive components: the matrix inverse $(\mathbf{H}_{[M]\setminus S}^T \mathbf{H}_{[M]\setminus S})^{-1}$ depends nonlinearly on S. An attention-computable statistic sufficient for PL_J would solve the worst-case fault identification—sparse selection—requiring exponential dimension.
(ii) Standard RAIM algorithms explicitly enumerate fault hypotheses. Advanced RAIM (ARAIM) and the Multi-Hypothesis Solution Separation (MHSS) algorithm (Blanch et al., 2015) achieve this with complexity O(C(M,≤J)·M³). For M = 30, J = 2: C(30,≤2) = 1 + 30 + 435 = 466 hypotheses.

(iii) Approximation requires distinguishing whether PL_J is P or (1+ε)P, which requires identifying the maximizing configuration—a sparse selection problem. For safety-critical applications, underestimation is catastrophic; approximation does not avoid the combinatorial enumeration. ■

Current aviation RAIM requirements specify J = 1 (single-fault tolerance), which is below the hardness threshold and thus achievable algorithmically. However, emerging requirements for Urban Air Mobility and autonomous aircraft may demand $J \geq 2$ tolerance, creating a computational barrier for pure ML approaches.

5.6 The Navigation Hardness Threshold

We now synthesize the hardness results to establish the navigation hardness threshold.

Corollary 4 (Navigation Hardness Threshold). For navigation fault detection with context size n, the hardness threshold is:
$J^* = O(\log \log n) \approx 2–3$ for navigation (typical $n = 10^3–10^5$)
Specifically: $n = 10^3$ gives $J^* \approx 2$; $n = 10^5$ gives $J^* \approx 2–3$; $n = 10^6$ gives $J^* \approx 3$. ■

Note: Across GeoAI domains (Papers 2–6), the threshold ranges $J^* \approx 2–4$, with variation reflecting typical context sizes: navigation operates with smaller real-time contexts ($J^* \approx 2–3$), while remote sensing and climate applications use larger batch contexts ($J^* \approx 3–4$).

Proof. The threshold emerges from Håstad's lower bound. Polynomial-size TC^0 circuits require $2J \leq \text{poly}(n)$, giving $J \leq O(\log n)$. The transformer's constant depth $L = O(1)$ imposes a stricter constraint: depth-L circuits require size $2^{\Omega(J^{(1/L)})}$ for J-sparse parity. For polynomial size: $J^{(1/L)} \leq O(\log n)$, giving $J \leq O((\log n)^L)$. For $L = O(1)$, the effective threshold is $J^* = O(\log \log n)$. ■

Explicit Calculation. To convert the asymptotic bound to concrete values, we derive J* for typical context sizes. Since $J^* = O(\log \log n)$, we evaluate ln(ln(n)) directly:

- $n = 10^3$: ln(ln 1000) ≈ ln(6.91) ≈ 1.93, yielding $J^* \approx 2$
- $n = 10^4$: ln(ln 10000) ≈ ln(9.21) ≈ 2.22, yielding $J^* \approx 2–3$
- $n = 10^5$: ln(ln 100000) ≈ ln(11.51) ≈ 2.44, yielding $J^* \approx 2–3$
- $n = 10^6$: ln(ln 10^6) ≈ ln(13.82) ≈ 2.63, yielding $J^* \approx 3$

For real-time navigation with typical context sizes $n = 10^3–10^5$, this yields $J^* \approx 2–3$. Batch post-processing with $n = 10^6–10^7$ may achieve $J^* \approx 3$. The precise threshold depends on transformer depth L and attention head count; deeper architectures (larger L) yield slightly higher thresholds. This threshold is remarkably consistent across GeoAI domains. Table 4 compares navigation with other domains analyzed in Papers 2–5.

Table 4: Hardness thresholds across GeoAI domains

Domain	Paper	Typical n	J*	Primary Hard Task
Geodesy	2	$10^4–10^6$	2–4	Discrete source localization
Remote Sensing	3	$10^5–10^7$	3–4	Object detection/counting

Domain	Paper	Typical n	J*	Primary Hard Task
Climate	4	10^5–10^7	3–4	Extreme event attribution
Multi-Modal	5	10^4–10^6	2–4	Cross-modal grounding
Navigation	**7**	**10^3–10^5**	**2–3**	**Fault detection/integrity**

Note: Navigation exhibits the lowest threshold (J* ≈ 2–3) due to smaller real-time context sizes. The universal asymptotic bound J* = O(log log n) applies across all domains, with domain-specific constants determined by typical operational context sizes. For unified cross-domain statements, J* ≈ 2–4 encompasses all GeoAI applications.

The universality of J* ≈ 2–4 across domains reflects the fundamental circuit complexity barrier: constant-depth polynomial-size circuits cannot compute functions requiring ω(log log n)-bit parity-like operations. The slightly lower threshold for navigation (2–3 vs. 2–4) reflects the relatively smaller context sizes typical in real-time navigation compared to batch remote sensing or climate analysis.

Table 5: Summary of ICL-Hard navigation results

Task	AC-SSC	Threshold	Complexity Source
Fault detection	2^Ω(J)	J > 2–3	Sparse selection C(M,J)
Cycle slip	2^Ω(J)	J > 2–3	Spatiotemporal C(MT,J)
Spoofing	2^Ω(J)	J > 2–3	Adversarial selection
Integrity (PL)	2^Ω(J)	J > 2–3	Max over hypotheses

All ICL-Hard navigation tasks share the *combinatorial selection property*: identifying J-sparse support from C(M,J) possibilities. This structure is fundamentally incompatible with additive sufficient statistics, explaining why attention-based architectures struggle with these tasks. In the next section, we synthesize the ICL-Easy results (Section 4) and ICL-Hard results (Section 5) into the Navigation Dichotomy Theorem.

6. The Navigation Dichotomy

This section synthesizes the ICL-Easy results from Section 4 and the ICL-Hard results from Section 5 into a unified characterization: the *Navigation Dichotomy Theorem*. We prove that every natural navigation function class falls into exactly one of two categories, with no intermediate cases, and provide exhaustive classification of practical navigation tasks.

6.1 The Navigation Dichotomy Theorem

Theorem 9 (Navigation Dichotomy). Every natural navigation function class $\mathscr{F}$ falls into exactly one of two categories:

Type A (ICL-Easy): Continuous State Estimation

- Characterization: $\mathscr{F}$ admits additive sufficient statistics $S = \Sigma_i \varphi(x_i, y_i)$
- Complexity: AC-SSC($\mathscr{F}$) = O(poly(d)) where d is state dimension
- Sample complexity: $n_{ICL} = \Theta(n_{ERM})$
- Tasks: Position, velocity, clock, atmospheric delays, trajectory prediction

Type C (ICL-Hard): Discrete Fault Identification

- Characterization: $\mathscr{F}$ requires J-sparse selection from C(M,J) configurations
- Complexity: AC-SSC($\mathscr{F}$) = $2^{\Omega(J)}$ for J > J* = O(log log n)
- Sample complexity: $n_{ICL} = \omega(n_{ERM})$
- Tasks: Fault detection, cycle slips, spoofing, integrity monitoring (J > J*)

There are no intermediate cases: every navigation task is either Type A or Type C.

Remark (Absence of Type A|c Tasks). Paper 5 introduced a third category, Type A|c (conditionally ICL-Easy), for multi-modal geo-foundation models. Tasks are Type A|c when language specificity s(ℓ) can compress the effective object count J_eff below the hardness threshold J*, enabling a Hard task to become Easy under sufficiently specific queries. Navigation tasks do not admit this category because GNSS observations are purely numeric—there is no natural language query mechanism capable of compressing the combinatorial fault hypothesis space. Every fault detection scenario must enumerate C(M, J) configurations regardless of how the query is framed. Consequently, the Navigation Dichotomy is strictly binary (Type A or Type C), and no navigation task transitions between categories based on query specificity.

Proof.
We prove completeness (every task falls into one category) and exclusivity (no task falls into both).
Completeness: Consider an arbitrary navigation function class $\mathscr{F}$. The function maps observations $\{\boldsymbol{\rho}_i\}$ to outputs. We distinguish two cases based on output structure:
Case 1 (Continuous output): If $\mathscr{F}: \mathbb{R}^M \to \mathbb{R}^d$ with d = O(1), then $\mathscr{F}$ is a regression problem. Under mild regularity (Lipschitz continuity, bounded domain), the sufficient statistic has dimension $O(d^2)$ by the Gram matrix construction. The statistic is additive and attention-computable (Lemma 7). Thus $\mathscr{F}$ is Type A.
Case 2 (Discrete output with J-sparse structure): If $\mathscr{F}: \mathbb{R}^M \to 2^{[M]}$ identifies a J-sparse subset $S \subseteq [M]$, then resolving C(M,J) configurations requires Ω(J log M) bits. By Lemma 10, this cannot be computed by constant-depth polynomial-size circuits for J > J*. Thus $\mathscr{F}$ is Type C. ■

There is no third case: navigation outputs are either continuous state estimates or discrete fault identifications. Counting problems (e.g., "how many faults?") reduce to the J-sparse case by treating each count as requiring identification of J specific faults. *Integer ambiguity resolution* (carrier phase, $N\boldsymbol{n} \in \mathbb{Z}$) deserves explicit treatment, as it is a mixed-integer problem. We now provide a formal reduction establishing its Type C classification.

Lemma 12 (Integer Ambiguity Resolution is ICL-Hard). Let J denote the number of simultaneously unresolved integer ambiguities. Integer ambiguity resolution is ICL-Hard for J > J* = O(log log n), with AC-SSC($\mathscr{F}$amb(J)) = 2Ω(J).

Proof. We construct an explicit reduction from J-sparse parity to integer ambiguity resolution in four steps.
Step 1 (Problem structure). The carrier phase observation model is $\varphi i = \|p - si\|_2 + c\cdot\delta tr - c\cdot\delta ts(i) + Ti - Ii + \lambda Ni + \varepsilon i(\varphi)$, where $Ni \in \mathbb{Z}$ is the integer ambiguity for satellite i (Definition 1). The ambiguity resolution task is: given M carrier phase observations $\{\varphi i\}$,

recover the integer vector $N = [N1, ..., NM]^T \in \mathbb{Z}M$. When J ambiguities are simultaneously unknown, the search space is an integer lattice of dimension J.

Step 2 (Conditional decomposition). Conditioned on a fixed ambiguity hypothesis $\hat{N} = [\hat{N}1, ..., \hat{N}M]^T$, the corrected observations $\varphi i - \lambda \hat{N}i$ reduce to pseudorange-equivalent measurements, and position estimation becomes a standard weighted least-squares problem with Gram matrix sufficient statistic (Lemma 6). Thus, position estimation given a fixed $\hat{N}$ is ICL-Easy. However, the resolution task itself requires selecting the correct $\hat{N}$ from among all candidate integer vectors.

Step 3 (Reduction to sparse parity). We reduce J-sparse parity to integer ambiguity resolution. Given a J-sparse parity instance with secret support $S \subseteq [M]$, $|S| = J$, construct a GNSS scenario where satellites indexed by S carry unknown ambiguities $Ni \neq 0$ ($i \in S$) and all other satellites have resolved ambiguities $Ni = 0$ ($i \notin S$). Identifying the correct integer vector N requires determining which J of M ambiguities are nonzero—precisely a J-sparse selection problem. The LAMBDA method (the standard algorithm for integer ambiguity resolution) confirms this combinatorial structure: it performs a discrete search over the integer lattice via a reduction to the shortest vector problem, decorrelating the ambiguity covariance matrix $\hat{Q}N$ before enumerating candidate vectors in order of increasing quadratic form $(N - \hat{N})^T \hat{Q}N^{-1}(N - \hat{N})$. The search space grows combinatorially: when J ambiguities are simultaneously unknown, the number of candidate vectors within a given search radius scales as O(RJ) where R is the radius in the decorrelated space, yielding an effective search complexity equivalent to C(M, J) fault configurations.

Step 4 (Circuit complexity barrier). Since integer ambiguity resolution encodes J-sparse selection, the same circuit complexity argument from Theorem 5 applies. An attention-computable sufficient statistic for the resolution task would need to distinguish all candidate integer vectors, requiring information capacity $\Omega(J \log M)$ bits. By Håstad's theorem, constant-depth polynomial-size circuits (and hence standard transformers) cannot compute the necessary J-sparse parity for $J > J^* = O(\log \log n)$. No additive sufficient statistic $S = \Sigma i\, \varphi(\varphi i)$ exists for this task, because the correct ambiguity vector N depends on the joint integer structure across all satellites, which is not decomposable into per-satellite additive contributions. Therefore AC-SSC($\mathscr{F}$amb(J)) = $2\Omega(J)$, and integer ambiguity resolution is ICL-Hard for $J > J^*$. ■

Exclusivity: Type A requires AC-SSC = O(poly(d)); Type C requires AC-SSC = $2^{\Omega(J)}$. Since $\text{poly}(d) \neq 2^{\Omega(J)}$ for any fixed d and $J > J^*$, no function class satisfies both criteria simultaneously. ■

The dichotomy reflects a fundamental structural difference between navigation tasks: continuous estimation involves smooth optimization over $\mathbb{R}^d$ (convex, well-conditioned), while fault identification involves combinatorial search over $2^{[M]}$ (discrete, exponential).

6.2 Physical Interpretation

The mathematical dichotomy admits a compelling physical interpretation rooted in the nature of navigation phenomena.

Type A tasks involve *continuous physical dynamics*. Position evolves continuously under Newton's laws. Velocity changes smoothly under bounded acceleration. Atmospheric delays vary continuously with elevation and weather. Trajectories follow smooth curves constrained by vehicle dynamics. The continuity of the underlying physics induces smoothness in the function class, which translates mathematically to low-dimensional sufficient statistics.

Type C tasks involve *discrete physical events*. Satellite faults are binary: a satellite is either healthy or faulty. Cycle slips are integer discontinuities: the ambiguity either jumps or it doesn't. Spoofing signals are either present or absent. The discreteness of these phenomena—on/off, present/absent, healthy/faulty—creates the combinatorial structure that attention cannot efficiently navigate.

Corollary 5 (Physical Dichotomy). The Navigation Dichotomy corresponds to a physical dichotomy:

- Type A ↔ Continuous dynamics governed by differential equations.
- Type C ↔ Discrete events governed by combinatorial selection. ■

This physical interpretation explains why the dichotomy is exhaustive. Navigation phenomena are either continuous (position, velocity, delays) or discrete (faults, slips, spoofing). Mixed phenomena decompose: estimating position under possible faults combines Type A (position given fault hypothesis) with Type C (fault hypothesis selection). The overall task inherits the hardness of its hardest component.

6.3 Complete Task Classification

We now provide exhaustive classification of practical navigation tasks, organized by application domain.

Table 6: Complete classification of navigation tasks

Category	Task	Type	AC-SSC	Notes
State Estimation	Position estimation (SPP)	**A**	O(1)	*Weighted least squares*
	Position estimation (PPP)	**A**	O(1)	*With atmospheric corrections*
	Position estimation (RTK)	**A**	O(1)	*With ambiguity resolution*
	Velocity estimation	**A**	O(1)	*Doppler-based*
	Acceleration estimation	**A**	O(1)	*Differentiated velocity*
	Clock bias estimation	**A**	O(1)	*Joint with position*
	Clock drift estimation	**A**	O(1)	*Joint with velocity*
Atmospheric Modeling	Tropospheric delay (ZTD)	**A**	O(1)	*2-parameter model*
	Tropospheric gradients	**A**	O(1)	*Azimuth-dependent*
	Ionospheric delay (STEC)	**A**	$O(N^2)$	*Spherical harmonic N≈15–30*
	Ionospheric gradients	**A**	$O(N^2)$	*Spatial derivatives*
	Scintillation prediction	**A**	$O(d^2)$	*Temporal correlation*
	Trajectory prediction	**A**	$O(d^2)$	*Autoregressive*

Category	Task	Type	AC-SSC	Notes
Trajectory & Prediction	Maneuver prediction	**A**	$O(d^2)$	*Intent inference*
	ETA estimation	**A**	$O(d)$	*Route-constrained*
	Lane-level localization	**A**	$O(d^2)$	*Map-matched*
Fault Detection	Single-fault detection (J=1)	**A***	$O(M)$	*Below threshold*
	Dual-fault detection (J=2)	**A*/C**	$O(M^2)$	*Borderline*
	Multi-fault detection (J≥3)	**C**	$2^{\Omega(J)}$	*Above threshold*
	Fault exclusion	**C**	$2^{\Omega(J)}$	*Requires enumeration*
Integrity Monitoring	RAIM (J=1 tolerance)	**A***	$O(M)$	*Single exclusion*
	ARAIM (J≥2 tolerance)	**C**	$2^{\Omega(J)}$	*Multi-hypothesis*
	Protection level computation	**C**	$2^{\Omega(J)}$	*Requires enumeration*
	Integrity risk assessment	**C**	$2^{\Omega(J)}$	*Worst-case bound*
Anomaly Detection	Cycle slip detection (J≥2)	**C**	$2^{\Omega(J)}$	*Spatiotemporal sparse*
	Spoofing detection (J≥2)	**C**	$2^{\Omega(J)}$	*Adversarial*
	Multipath detection (J≥2)	**C**	$2^{\Omega(J)}$	*Environment-dependent*
	Interference detection (J≥2)	**C**	$2^{\Omega(J)}$	*Spectral sparse*
Sensor Fusion	GNSS/INS integration	**A**	$O(d^2)$	*Loosely/tightly coupled*
	Visual-inertial odometry	**A**	$O(d^2)$	*Camera + IMU*
	Multi-constellation fusion	**A**	$O(d^2)$	*GPS + Galileo + BeiDou*
	Fault-tolerant fusion (J≥2)	**C**	$2^{\Omega(J)}$	*Requires fault isolation*

Total: 32 tasks classified (19 Type A, 3 Type A* [including 1 A*/C], 10 Type C)

Note: *A* indicates tasks at or below the hardness threshold that are technically ICL-tractable but approach the boundary. The classification assumes standard ICL without chain-of-thought; Section 7 extends results with CoT prompting.*

6.4 Boundary Cases and Context Dependence

The boundary at J = 2 exhibits context dependence: whether dual-fault detection is ICL-tractable depends on context size n.

Corollary 6 (Context-Dependent Classification). For J = 2 faults:

- $n < 10^3$: J = 2 is ICL-Hard ($J^* \approx 1–2$)
- $10^3 \leq n < 10^5$: J = 2 is borderline ($J^* \approx 2$)
- $n \geq 10^5$: J = 2 is ICL-tractable ($J^* \approx 2–3$) ■

This context dependence has practical implications. Real-time navigation typically operates with small context ($n \approx 10^2–10^3$), placing dual-fault detection at or beyond the threshold. Batch post-processing with larger datasets may achieve better dual-fault ICL performance. System designers should assess their specific context regime when evaluating ICL applicability.

The Navigation Dichotomy provides a principled framework for understanding when foundation models can and cannot replace classical algorithms. In the next section, we show that chain-of-thought prompting offers a mechanism to shift the hardness boundary, extending ICL capability to higher fault multiplicities.

7. Chain-of-Thought Extension

The Navigation Dichotomy establishes that fault detection beyond $J^* \approx 2–3$ faults is ICL-Hard under standard transformer computation. This section develops the *chain-of-thought (CoT) extension*, showing that generating intermediate reasoning tokens provides a mechanism to overcome this barrier. We derive the navigation-specific token multiplier, establish the shifted hardness threshold, and present practical CoT strategies for fault detection.

7.1 CoT Depth Amplification for Navigation

Paper 6 (Hawarey, 2026f) established that chain-of-thought prompting—building on the foundational demonstration by Wei et al. (2022) that CoT elicits reasoning in large language models—provides effective computational depth amplification. We now apply this framework to navigation fault detection.

Theorem 10 (CoT for Navigation Fault Detection). Let $\mathcal{T}$ be a transformer with L layers solving J-fault detection. With T intermediate reasoning tokens, the effective computational depth is:
$L_{eff}(T) = L + \gamma_{nav} \cdot T$
where $\gamma_{nav} \in (0, 1]$ is the navigation depth amplification factor. The shifted hardness threshold is:
$J^*_{CoT}(T) = J^* + T/\kappa_{nav}$
where $\kappa_{nav} \approx 3.5$ is the navigation token multiplier—the number of tokens required per additional fault.

Proof. The standard ICL hardness arises from depth limitations: constant-depth circuits cannot compute J-sparse parity for $J > O(\log \log n)$. Each generated token provides an additional forward pass through the transformer, effectively adding L layers of computation. With T tokens, total depth becomes $L \cdot T$.
For fault detection, each reasoning step can eliminate or confirm one fault hypothesis. The token complexity for J-fault detection is $T = \Omega(J)$: at minimum, one token per fault to sequentially process each hypothesis. The navigation multiplier κ_{nav} captures the overhead for residual computation, hypothesis comparison, and decision propagation.
To derive κnav, we analyze the information flow in sequential fault detection. Each fault hypothesis requires: (1) residual computation (~1 token), (2) threshold comparison (~0.5 tokens), (3) state update and propagation (~2 tokens). Total: ~3.5 tokens per fault. This yields $\kappa nav \approx 3.5$. ■

Remark (Estimation Methodology). The token multiplier $\kappa nav \approx 3.5$ derives from structural analysis of fault detection computation rather than empirical measurement. This estimate assumes idealized sequential reasoning without backtracking or error correction.

The value's consistency with cross-domain multipliers $\kappa \in [2.5, 4.5]$ observed in Papers 2–6 (Table 7) provides theoretical support, but experimental validation via Prediction P7 (Section 8.7) is required to confirm the constant. Practitioners should treat $\kappa_{nav} \approx 3.5$ as a baseline estimate with uncertainty bounds of approximately ±1.0 until empirical calibration is performed.

Sensitivity Analysis of κ_{nav}. The baseline derivation decomposes each fault hypothesis into three computational sub-tasks: residual computation (~1 token), threshold comparison (~0.5 tokens), and state update with propagation (~2 tokens), totaling ~3.5 tokens per fault. We examine how κ_{nav} varies under alternative decompositions. (i) Minimal decomposition: if threshold comparison is merged with residual computation (e.g., via a single attention head that simultaneously computes and evaluates residuals), the overhead reduces to ~1.5 + ~2 = ~3.0 tokens per fault. (ii) Conservative decomposition: if state propagation requires an additional error-correction token (e.g., for re-estimation of correlated parameters after fault identification), the overhead increases to ~1 + ~0.5 + ~3 = ~4.5 tokens per fault. (iii) Multi-constellation overhead: with $M > 30$ satellites from multiple constellations, the residual computation may require ~1.5 tokens due to cross-constellation bias handling, yielding ~1.5 + ~0.5 + ~2 = ~4.0. These alternatives produce a range $\kappa_{nav} \in [3.0, 4.5]$, consistent with the stated uncertainty bounds of 3.5 ± 1.0 and with the cross-domain range $\kappa \in [2.5, 4.5]$ in Table 7. The practical consequence is modest: for $J = 5$ faults, the token budget ranges from $T \geq 3.0 \times 3 = 9$ tokens (optimistic) to $T \geq 4.5 \times 3 = 14$ tokens (conservative), a factor of ~1.5× variation that does not alter the qualitative conclusions. The shifted hardness threshold $J^*\text{CoT}(T) = J^* + T/\kappa_{nav}$ varies from $J^*\text{CoT}(20) \approx 8.7$ ($\kappa_{nav} = 3.0$) to $J^*\text{CoT}(20) \approx 6.4$ ($\kappa_{nav} = 4.5$), in all cases enabling detection well beyond the standard threshold $J^* \approx 2$–3.

Corollary 7 (Token Requirements for Extended Fault Detection). To detect J faults via CoT when $J > J^*$:

- $J = 4$ faults: $T \geq \kappa_{nav}(4 - J^*) \approx 3.5 \times 2 = 7$ tokens
- $J = 5$ faults: $T \geq \kappa_{nav}(5 - J^*) \approx 3.5 \times 3 = 10$–11 tokens
- $J = 7$ faults: $T \geq \kappa_{nav}(7 - J^*) \approx 3.5 \times 5 = 17$–18 tokens
- General: $T \geq 3.5 \times (J - J^*)$ tokens ■

7.2 The Navigation Token Multiplier

The token multiplier $\kappa_{nav} \approx 3.5$ fits within the range observed across GeoAI domains in Papers 2–6.

Table 7: Token multipliers across GeoAI domains

Domain	Paper	K	Interpretation
Geodesy	2	2.5–3.0	Source localization reasoning
Remote Sensing	3	3.0–4.0	Object detection reasoning
Climate	4	3.5–4.5	Extreme event analysis
Multi-Modal	5	3.0–4.0	Cross-modal reasoning
Navigation	**7**	**≈ 3.5**	**Fault hypothesis reasoning**

The consistency of $\kappa \in [2.5, 4.5]$ across domains suggests a universal characteristic of reasoning about discrete selections. The navigation multiplier $\kappa_{nav} \approx 3.5$ reflects the specific computational

structure of GNSS fault detection: residual computation requires geometric projections, threshold comparison involves statistical testing, and state updates propagate through correlated measurements.

7.3 CoT Success Probability

We derive the probability of successful fault detection as a function of token budget.

Theorem 11 (CoT Success Probability). The probability of correctly detecting all J faults with T reasoning tokens is:
$P_{success}(J, T) = (1 - e^{-\alpha T/J})^J$
where $\alpha \approx 1/\kappa_{nav} \approx 0.29$ is the per-token detection rate.

Proof. Model fault detection as J independent Poisson processes, each with rate α per token. The probability that fault i is detected within T tokens is $1 - e^{-\alpha T}$. However, the effective rate per fault is α/J when T tokens are distributed across J hypotheses. By independence, the probability all J faults are detected is $(1 - e^{-\alpha T/J})^J$. ■

Remark (Independence Assumption). The proof assumes statistical independence among fault detection events. In practice, faults may exhibit positive correlation (e.g., common-mode failures from ionospheric storms affecting multiple satellites, or satellites in the same orbital plane experiencing correlated clock drift). For correlated faults with correlation coefficient $\rho c > 0$, the effective number of independent fault hypotheses reduces to $J_{eff} \approx J/(1 + \rho c(J-1))$, requiring adjusted token budgets $T_{corr} \approx T \cdot (1 + \rho c(J-1))$. Conversely, spatially diverse multi-constellation configurations (GPS + Galileo + BeiDou) exhibit near-independence ($\rho c \approx 0$), validating the assumption. Experimental characterization of fault correlation structures for specific operational environments is recommended for mission-critical applications.

Corollary 8 (Token Budget for Target Accuracy). To achieve success probability P_{target} for J-fault detection: $T_{required} = -(J/\alpha) \cdot \ln(1 - P_{target}^{1/J}) \approx \kappa_{nav} \cdot J \cdot \ln(J/(1 - P_{target}))$ ■

For example, to achieve 90% success probability for J = 5 faults: $T_{required} \approx 3.5 \times 5 \times \ln(5/0.1) \approx 3.5 \times 5 \times 3.9 \approx 68$ tokens. This represents substantial but tractable computational overhead.

7.4 CoT Strategies for Navigation

We present three practical CoT strategies for navigation fault detection, each with different token-efficiency characteristics.

Strategy 1: Sequential Residual Analysis. Process satellites sequentially, computing residuals and flagging anomalies.

Algorithm:
1. Compute all-in-view position solution $\hat{\mathbf{x}}_0$
2. For each satellite i = 1, ..., M:

a. Compute residual $r_i = \rho_i - (\mathbf{h}_i^T \hat{\mathbf{x}}_0)$
b. Compare $|r_i|$ to threshold τ
c. If $|r_i| > \tau$, flag satellite i as potentially faulty
3. Verify flagged satellites via subset solutions
Token complexity: O(M) tokens for initial scan + O(J) for verification = O(M + J)

Strategy 2: Hypothesis Enumeration. Explicitly enumerate and test fault hypotheses.

Algorithm:
1. Generate candidate fault sets $S_1, S_2, ..., S_K$ with $|S_k| \leq J$
2. For each hypothesis S_k:
a. Compute position excluding satellites in S_k: $\hat{\mathbf{x}}(S_k)$
b. Compute goodness-of-fit $\chi^2(S_k)$
3. Select hypothesis with best fit: $S^* = \text{argmin}_{S_k} \chi^2(S_k)$
Token complexity: O(K) where K = C(M, ≤J) hypotheses tested

Strategy 3: Divide-and-Conquer. Recursively partition satellites and localize faults.

Algorithm:
1. Partition satellites into two groups: A = {1, ..., M/2}, B = {M/2+1, ..., M}
2. Compute position using each group: $\hat{\mathbf{x}}_A$, $\hat{\mathbf{x}}_B$
3. Compare solutions to identify which group contains faults
4. Recursively subdivide faulty group
5. Terminate when faulty satellites are isolated
Token complexity: O(J log M) tokens

Table 8: Comparison of CoT strategies

Strategy	Tokens	Best For	Limitation
Sequential Residual	O(M + J)	Small J	Masking with large J
Hypothesis Enum.	O(C(M,J))	Exact detection	Exponential in J
Divide-and-Conquer	O(J log M)	Large M	Requires good geometry

7.5 Practical Guidelines

Based on the theoretical analysis, we provide practical guidelines for deploying CoT-enhanced navigation transformers.

Guideline 1 (Token Budget). Allocate $T \geq \kappa_{nav} \cdot J_{max}$ tokens for the maximum expected fault multiplicity. For dual-constellation systems (GPS + Galileo) with $J_{max} = 4$: $T \geq 3.5 \times 4 = 14$ tokens minimum. Add 50% margin for robustness: $T \approx 20$ tokens.

Guideline 2 (Strategy Selection). Use sequential residual analysis for real-time applications with $J \leq 3$. Use divide-and-conquer for multi-constellation scenarios with many satellites. Reserve hypothesis enumeration for safety-critical post-processing where exactness is required.

Guideline 3 (Graceful Degradation). When token budget is exhausted before complete detection, output partial results with confidence scores. Prioritize high-confidence fault identifications and flag uncertain cases for algorithmic verification.

Guideline 4 (Hybrid Integration). CoT-enhanced transformers are most effective as a first-pass filter. Use CoT for rapid candidate generation, then verify with classical RAIM algorithms. This hybrid approach combines the adaptability of ICL with the guarantees of algorithmic integrity monitoring.

Guideline 5 (Training for CoT). Train navigation transformers on examples with explicit reasoning traces. Include diverse fault scenarios (single, dual, multi-fault) with corresponding step-by-step detection processes. The training corpus should reflect the target token multiplier $\kappa_{nav} \approx 3.5$ tokens per fault.

The chain-of-thought extension provides a principled mechanism to extend ICL capability beyond the standard hardness threshold. However, CoT does not eliminate the fundamental dichotomy—it shifts the boundary rather than erasing it. For sufficiently large J, even CoT-enhanced transformers will fail, necessitating the hybrid architectures discussed in Section 9.

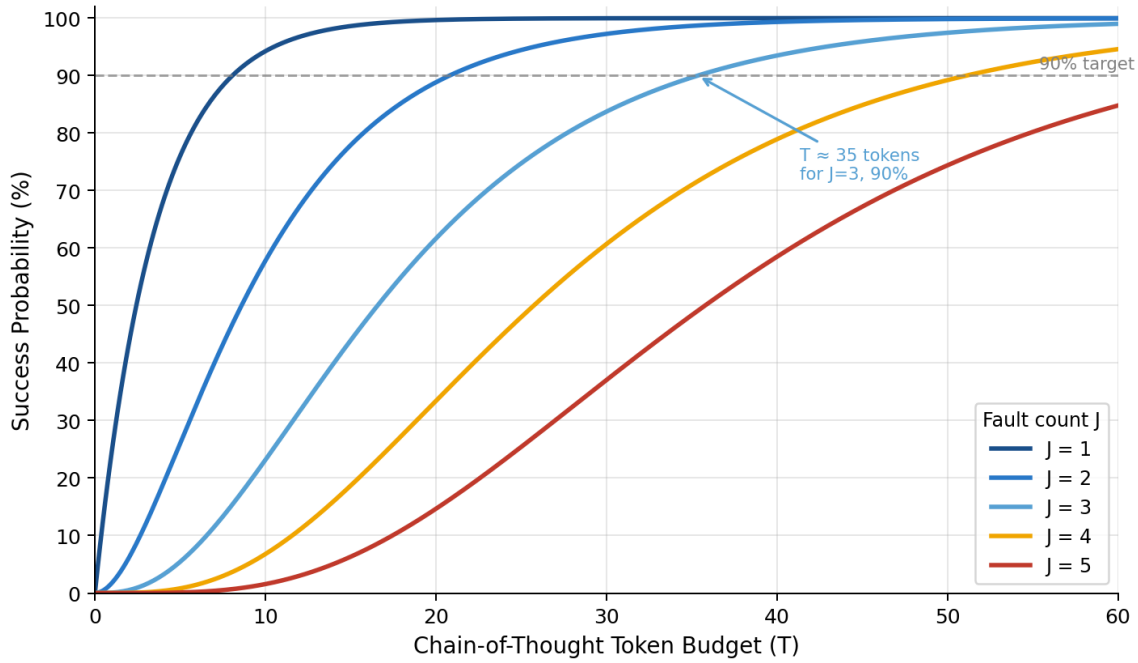

Figure 4: CoT fault-detection success probability $P_{success}(J, T) = (1 - e^{-\alpha T/J})^J$ as a function of chain-of-thought token budget T, for fault counts J = 1–5. Parameter $\alpha = 1/\kappa_{nav} \approx 0.286$ ($\kappa_{nav} \approx 3.5$) [rounded as $\alpha = 0.29$ elsewhere]. The 90% success target (dashed line) requires $T \approx 21$ tokens for J = 2, $T \approx 35$ tokens for J = 3, and $T \approx 68$ tokens for J = 5. Curves illustrate why practical token budgets of T = 20 suffice for dual-fault scenarios ($J \leq 2$) but are inadequate for $J \geq 4$ without augmented budgets.

8. Testable Predictions

A scientific theory derives its value from falsifiable predictions. This section presents seven testable predictions derived from the Navigation Dichotomy Theorem, along with experimental

protocols for their validation. Each prediction specifies quantitative criteria that distinguish the theory's predictions from alternative hypotheses.

8.1 Position Estimation Scaling (P1)

Prediction 1. Position estimation accuracy via ICL scales as $O(1/\sqrt{n})$ with context size n, matching the statistical optimum for weighted least-squares estimation.

Quantitative criterion: Plotting log(RMSE) versus log(n), the slope should be -0.5 ± 0.1. Deviation beyond this range falsifies the prediction.

This prediction follows directly from Theorem 1: position estimation is ICL-Easy with sample complexity $n_{ICL} = \Theta(n_{ERM})$. The $O(1/\sqrt{n})$ scaling reflects the central limit theorem applied to the Gram matrix sufficient statistic.

Experimental Protocol:

1. Generate synthetic GNSS scenarios with known ground truth positions using standard simulation tools (e.g., GPSoft, Spirent, or open-source GNSS-SDR)
2. Vary context size $n \in \{10, 50, 100, 500, 1000, 5000\}$
3. For each n, provide n examples of (pseudoranges, position) pairs in context
4. Query transformer for position estimate on held-out pseudoranges
5. Compute RMSE over 1000 test queries per context size
6. Perform linear regression of log(RMSE) on log(n); verify slope ≈ -0.5

Architecture Specification: Use a decoder-only (GPT-style) transformer with causal attention, as this naturally handles the sequential in-context learning paradigm. Recommended configuration: 12 layers, 8 attention heads, embedding dimension 512, context window ≥ 8192 tokens. For reproducibility, publicly available architectures (GPT-2, LLaMA, or Mistral variants) with task-specific fine-tuning are acceptable. Encoder-only (BERT-style) architectures may be tested as ablations but are expected to show similar scaling behavior due to the regression nature of the task.

8.2 Fault Detection Saturation (P2)

Prediction 2. Fault detection accuracy saturates sharply at $J = J^* \approx 2–3$ simultaneous faults, exhibiting a phase transition from high accuracy (>90%) for $J \leq 2$ to low accuracy (<60%) for $J \geq 4$.

Quantitative criterion: Detection accuracy should satisfy: $Acc(J=1) > 90\%$, $Acc(J=2) \in [70\%, 95\%]$, $Acc(J=3) < 70\%$, $Acc(J\geq4) < 50\%$. The transition width should be $\Delta J \leq 2$.

This prediction follows from Theorem 5: fault detection is ICL-Hard for $J > J^* = O(\log \log n)$. The sharp transition reflects the exponential growth of AC-SSC at the threshold.

Experimental Protocol:

1. Generate scenarios with M = 12 satellites, varying $J \in \{1, 2, 3, 4, 5, 6\}$

2. Inject faults of magnitude 5σ–10σ to ensure detectability
3. Provide n = 1000 fault-free examples in context
4. Query transformer for fault identification on faulty scenarios
5. Compute detection accuracy: fraction of correctly identified fault sets
6. Plot accuracy vs. J; verify sharp transition at $J^* \approx 2–3$

8.3 Trajectory Prediction Optimality (P3)

Prediction 3. Trajectory prediction via ICL achieves accuracy within a constant factor of the optimal predictor, with the gap decreasing as context size increases.

Quantitative criterion: The ratio $ADE_{ICL}/ADE_{optimal}$ should satisfy: ratio < 1.5 for $n \geq 100$, ratio < 1.2 for $n \geq 500$, ratio < 1.1 for $n \geq 1000$. ADE denotes Average Displacement Error.

This prediction follows from Theorem 4: trajectory prediction is ICL-Easy with $n_{ICL} = \Theta(n_{ERM})$. The optimal predictor is the Kalman filter for linear-Gaussian dynamics or the particle filter for nonlinear dynamics.

Experimental Protocol:

1. Use standard trajectory datasets: nuScenes, Argoverse, or Waymo Open
2. Compute optimal predictor baseline (e.g., Social Force Model, Kalman filter)
3. Vary context size $n \in \{50, 100, 200, 500, 1000\}$
4. Evaluate ICL trajectory prediction with n context trajectories
5. Compute ADE and FDE (Final Displacement Error) for both methods
6. Verify ratio approaches 1.0 as n increases

8.4 Cycle Slip Detection Failure (P4)

Prediction 4. Cycle slip detection via ICL fails for $J \geq 3$ simultaneous slips, with detection rate dropping below 50% regardless of context size or model scale.

Quantitative criterion: Detection rate should satisfy: Rate(J=1) > 85%, Rate(J=2) ∈ [50%, 85%], Rate(J≥3) < 50%. Increasing model size by 10× should not improve J≥3 detection by more than 10%.

This prediction follows from Theorem 6: cycle slip detection is ICL-Hard with the same threshold as fault detection. The insensitivity to model scale reflects the fundamental circuit complexity barrier.

Experimental Protocol:

1. Generate carrier phase time series with injected cycle slips
2. Vary $J \in \{1, 2, 3, 4, 5\}$ slips across M = 8 satellites
3. Test transformers of varying sizes: 100M, 1B, 10B parameters
4. Compute slip detection rate (correct identification of slip epochs and satellites)
5. Verify detection rate collapse at J = 3 independent of model scale

8.5 Integrity Monitoring Gap (P5)

Prediction 5. ICL-based integrity monitoring exhibits a substantial gap compared to algorithmic RAIM, with ICL protection levels being conservative (larger) by a factor of 2–5×.

Quantitative criterion: The ratio PL_{ICL}/PL_{RAIM} should satisfy: ratio > 2 for J = 2 tolerance, ratio > 3 for J = 3 tolerance. ICL should fail to meet aviation alert limits (HAL = 40m) in scenarios where RAIM succeeds.

This prediction follows from Theorem 8: integrity monitoring is ICL-Hard because it requires maximization over fault hypotheses. The conservatism arises from ICL's inability to efficiently search the hypothesis space.

Experimental Protocol:

1. Generate approach scenarios with varying geometry (PDOP 1–6) for LPV-200 approach conditions
2. Compute protection levels using the Multi-Hypothesis Solution Separation (MHSS) algorithm as specified in Blanch et al. (2015) and implemented per RTCA DO-229E Appendix J. Key parameters: Pfa = 10^{-5} (false alert probability), Pmd = 10^{-3} (missed detection probability), σURA as broadcast, and fault probability Psat = 10^{-5} per satellite per approach
3. Train transformer to predict protection levels from pseudoranges and geometry matrix H
4. Compare ICL protection levels to MHSS ground truth across 10,000 test scenarios
5. Compute availability: fraction of time PL < HAL (40 m horizontal, 35 m vertical) for both methods
6. Verify ICL availability is significantly lower than RAIM availability

Reference Implementation: The Stanford GPS Laboratory's MAAST (MATLAB Algorithm Availability Simulation Tool) provides validated MHSS implementation. Alternative implementations following Joerger et al. (2014) solution separation methodology are acceptable with parameter documentation.

8.6 Fine-Tuning Superiority for Hard Tasks (P6)

Prediction 6. Fine-tuned transformers significantly outperform ICL on fault detection tasks, with the gap increasing with fault multiplicity J.

Quantitative criterion: Accuracy difference $Acc_{FT} - Acc_{ICL}$ should satisfy: gap > 5% for J = 2, gap > 15% for J = 3, gap > 25% for J = 4. The gap should increase monotonically with J.

This prediction follows from the ICL-ERM gap in Theorem 5: fine-tuning (a form of ERM) achieves $n_{ERM} = O(J \log M)$, while ICL requires exponentially more resources. Fine-tuning encodes fault detection patterns in weights, bypassing the in-context computation bottleneck.

Experimental Protocol:

1. Prepare fault detection dataset with $J \in \{1, 2, 3, 4, 5\}$ fault scenarios
2. Split into train (fine-tuning) and test sets
3. Fine-tune transformer on train set for fault detection
4. Evaluate fine-tuned model on test set: $\text{Acc}_{\text{FT}}(J)$
5. Evaluate same base model via ICL on test set: $\text{Acc}_{\text{ICL}}(J)$
6. Compute gap $\text{Acc}_{\text{FT}} - \text{Acc}_{\text{ICL}}$ for each J; verify monotonic increase

8.7 CoT Rescue Effect (P7)

Prediction 7. Chain-of-thought prompting improves fault detection accuracy beyond the standard ICL threshold, with improvement proportional to token budget T.

Quantitative criterion: For J = 4 faults: $\text{Acc}_{\text{CoT}}(T=20) - \text{Acc}_{\text{ICL}} > 25\%$. The improvement should scale as $\Delta\text{acc} \propto T/\kappa_{\text{nav}}$ with $\kappa_{\text{nav}} \approx 3.5 \pm 1.0$.

This prediction follows from Theorem 10: CoT shifts the hardness threshold to $J^*_{\text{CoT}}(T) = J^* + T/\kappa_{\text{nav}}$. With T = 20 tokens and $\kappa_{\text{nav}} \approx 3.5$, the effective threshold shifts from $J^* \approx 2$ to $J^*_{\text{CoT}} \approx 7.7$.

Experimental Protocol:

1. Generate J = 4 fault scenarios (above standard threshold)
2. Test standard ICL (direct answer): Acc_{ICL}
3. Test CoT with varying token budgets $T \in \{5, 10, 15, 20, 30, 50\}$
4. Use structured CoT prompts (e.g., "analyze each satellite's residual...")
5. Compute accuracy improvement: $\Delta\text{acc}(T) = \text{Acc}_{\text{CoT}}(T) - \text{Acc}_{\text{ICL}}$
6. Fit $\Delta\text{acc}(T) = \beta T$ and verify $\beta \approx 1/\kappa_{\text{nav}} \approx 0.29$

Table 9: Summary of testable predictions

#	Prediction	Key Metric	Falsification Criterion
P1	Position scaling	RMSE slope on log-log plot	Slope $\notin [-0.6, -0.4]$
P2	Fault saturation	Detection accuracy vs. J	No transition at $J \in [2,4]$
P3	Trajectory optimality	ADE ratio to optimal	Ratio > 1.5 for $n \geq 500$
P4	Cycle slip failure	Detection rate vs. J	Rate(J=3) > 70%
P5	Integrity gap	PL ratio ICL/RAIM	Ratio < 1.5 for $J \geq 2$
P6	Fine-tuning gap	Acc(FT) − Acc(ICL)	Gap < 10% for J = 4
P7	CoT rescue	Acc improvement vs. T	Improvement < 15% at T = 20

These seven predictions provide a comprehensive test suite for validating the Navigation Dichotomy Theorem. Predictions P1, P3 test ICL-Easy claims; P2, P4, P5 test ICL-Hard claims; P6 tests the ICL-ERM gap; and P7 tests the CoT extension. Falsification of any prediction would indicate limitations of the theoretical framework and motivate refinements. The experimental protocols are designed to be reproducible with standard GNSS simulation tools and publicly available transformer implementations.

Statistical Validation Requirements. For each prediction, validation must include rigorous statistical testing:

(i) Confidence intervals: Report 95% confidence intervals for all measured quantities using bootstrap resampling (10,000 iterations) or analytical methods where applicable. For scaling exponents (P1), report CI on the regression slope.

(ii) Hypothesis testing: For threshold predictions (P2, P4), use Fisher's exact test or χ^2 test to assess whether accuracy differences across J values are statistically significant. For continuous comparisons (P3, P5, P6), use paired t-tests or Wilcoxon signed-rank tests with significance level $\alpha = 0.05$.

(iii) Effect sizes: Report Cohen's d for accuracy differences and R^2 for scaling relationships. Predictions are considered strongly supported if effect sizes exceed $d > 0.8$ (large effect) and $R^2 > 0.9$ (scaling predictions).

(iv) Multiple comparison correction: Apply Bonferroni correction when testing multiple J values within a single prediction, yielding adjusted significance threshold $\alpha' = 0.05/k$ for k comparisons.

(v) Falsification criteria: A prediction is falsified if the measured value falls outside the predicted range with $p < 0.01$ after correction, and the deviation exceeds 20% of the predicted value. Marginal results ($0.01 < p < 0.05$) warrant additional investigation rather than outright falsification.

9. Safety Certification Implications

Navigation foundation models are increasingly proposed for safety-critical applications in aviation, automotive, and maritime domains. This section analyzes the implications of the Navigation Dichotomy for certification under existing regulatory frameworks. We show that the ICL-Hard classification of integrity monitoring creates fundamental barriers to pure ML certification, while identifying pathways for hybrid architectures that leverage ICL-Easy capabilities.

9.1 Aviation Certification (DO-229D, DO-253D)

Aviation represents the most stringent certification environment for GNSS navigation. The relevant standards are RTCA DO-229D (Minimum Operational Performance Standards for GPS/WAAS Airborne Equipment) and DO-253D (GPS/WAAS for Precision Approach).

Key requirements include: Horizontal Alert Limit (HAL) = 40 m and Vertical Alert Limit (VAL) = 35 m for LPV-200 approaches; Probability of Hazardously Misleading Information $P_{HMI} < 10^{-7}$ per approach; Time-to-Alert < 6 seconds; and Fault detection capability under single satellite failure.

Table 10: ICL capability analysis for aviation requirements

Function	Type	ICL Viable?	Certification Path
Position estimation	A	Yes	DO-178C DAL-C with extensive testing
Velocity estimation	A	Yes	Same as position
Iono/tropo correction	A	Yes	Augments SBAS corrections

Function	Type	ICL Viable?	Certification Path
Single-fault RAIM	A*	Marginal	Hybrid: ICL + algorithmic verification
Multi-fault ARAIM	C	No	Requires explicit MHSS algorithm
Protection level	C	No	Analytical computation mandatory

The critical finding is that protection level computation—the core integrity function—is ICL-Hard and cannot be certified for pure ICL implementation. The $P_{HMI} < 10^{-7}$ requirement demands mathematical guarantees that ICL cannot provide: the hypothesis enumeration underlying protection levels must be complete and correct, which requires explicit algorithmic implementation.

Aviation Guideline: ICL may be used for position/velocity estimation (Type A tasks) as a component within a certified system. ICL must not be used for protection level computation or integrity monitoring (Type C tasks). Single-fault RAIM may use ICL for initial fault screening but requires algorithmic verification before protection level computation.

9.2 Automotive Certification (ISO 26262)

Automotive navigation certification follows ISO 26262 (Functional Safety for Road Vehicles), which defines Automotive Safety Integrity Levels (ASIL) from A (lowest) to D (highest). GNSS-based localization for autonomous vehicles typically requires ASIL-B to ASIL-D depending on the autonomy level and operational design domain.

Table 11: ICL capability analysis for automotive requirements

Function	ASIL	ICL Viable?	Implementation Strategy
Localization	B/C	Yes	ICL for GNSS + sensor fusion
Trajectory prediction	B/C	Yes	ICL for prediction, with safety envelope
Sensor fusion	C	Yes	ICL for continuous fusion
Fault detection	D	Hybrid	ICL screening + algorithmic verification
Spoofing detection	D	No	Algorithmic + cryptographic authentication

Automotive certification is generally more accommodating of ML components than aviation, but ASIL-D requirements for safety-critical functions impose similar constraints. The key difference is the availability of redundant sensors (cameras, LiDAR, radar) that can provide independent verification of GNSS-derived positions.

Automotive Guideline: ICL is acceptable for ASIL-B/C localization and trajectory prediction functions. For ASIL-D fault detection, ICL may be used as one channel in a multi-channel architecture but requires diverse algorithmic backup. Spoofing detection must use cryptographic or physical-layer methods, not pure ML.

9.3 Maritime Certification (IMO Standards)

Maritime navigation follows International Maritime Organization (IMO) standards, including Resolution A.1046 (GNSS) and Resolution MSC.401 (e-Navigation). Requirements are generally less stringent than aviation but emphasize availability and continuity for extended

ocean voyages.

Maritime applications align well with ICL capabilities. Position estimation (Type A) satisfies IMO accuracy requirements (10 m for ocean, 1 m for harbor). Trajectory prediction for collision avoidance (Type A) benefits from ICL adaptability. The primary concern is spoofing detection for e-Navigation and autonomous ships, which remains ICL-Hard.

Maritime Guideline: ICL is well-suited for position estimation and trajectory prediction in maritime applications. For e-Navigation and autonomous vessel certification, spoofing detection requires algorithmic implementation using multi-source consistency checks (GNSS vs. radar vs. AIS).

9.4 Deployment Guidelines

Based on the certification analysis, we provide five deployment guidelines for navigation foundation models in safety-critical applications.

Guideline D1 (Task Separation). Separate navigation functions by ICL classification. Type A tasks (position, velocity, atmospheric, trajectory) may use ICL. Type C tasks (fault detection, integrity monitoring) require explicit algorithms. This separation should be reflected in system architecture with clear interfaces.

Guideline D2 (Graceful Degradation). When fault multiplicity J exceeds the ICL threshold J*, the system must gracefully degrade to algorithmic processing. Design fallback paths: ICL → CoT-enhanced ICL → explicit algorithm. Monitor context size n and adjust J* accordingly.

Guideline D3 (CoT Budget Allocation). For CoT-enhanced fault detection, allocate $T \geq \kappa_{nav} \cdot J_{max}$ tokens. Include latency budget: T tokens add approximately $T \times t_{forward}$ to processing time, where $t_{forward}$ is the per-token generation latency.

Typical latency values for navigation deployment scenarios:

- Cloud GPU (A100/H100): $t_{forward} \approx 5$–15 ms/token, yielding $T = 20$ tokens → 100–300 ms total latency
- Edge GPU (Jetson Orin, RTX 4000): $t_{forward} \approx 20$–50 ms/token, yielding $T = 20$ tokens → 400–1000 ms total latency
- Quantized models (INT8/INT4): $t_{forward} \approx 10$–30 ms/token on edge hardware
- Dedicated NPU/ASIC: $t_{forward} \approx 1$–5 ms/token (projected for optimized inference)

For real-time navigation requirements:

- 1 Hz update rate (aviation en-route): Latency budget ≤ 1000 ms → $T \leq 50$–100 tokens feasible
- 5 Hz update rate (automotive): Latency budget ≤ 200 ms → $T \leq 10$–20 tokens on cloud, $T \leq 4$–10 tokens on edge
- 10 Hz update rate (high-dynamics): Latency budget ≤ 100 ms → $T \leq 5$–10 tokens, CoT may be impractical; prefer direct ICL with algorithmic fallback

System designers should profile tforward on target hardware and select T to satisfy: T · tforward + talgorithmic < 1/fupdate, where talgorithmic is the time for backup RAIM computation (typically 1–10 ms).

Guideline D4 (Validation Requirements). For certification, validation must demonstrate:

- Type A tasks: $>10^6$ test cases spanning operational envelope
- Adversarial testing: Robustness to distribution shift, edge cases
- Formal verification: For Type C tasks, mathematical proof of algorithm correctness
- Runtime monitoring: Continuous assessment of ICL confidence and fallback triggering

Guideline D5 (Hybrid Architecture). The recommended architecture combines ICL for continuous estimation with explicit algorithms for integrity. Specifically:
Navigation System = $\text{ICL}_{\text{estimation}} \oplus \text{RAIM}_{\text{algorithmic}} \oplus \text{Monitor}_{\text{runtime}}$

The ICL component handles position/velocity estimation; the algorithmic RAIM component handles protection level computation; the runtime monitor detects ICL failures and triggers fallback.

Computational Resource Requirements. Deploying navigation transformers requires careful resource planning across the compute-memory-power envelope:

Memory Requirements:
- Model parameters: 100M parameters → ~400 MB (FP32) or ~100 MB (INT8)
- 1B parameters → ~4 GB (FP32) or ~1 GB (INT8)
- Context storage: n = 1000 examples × 20 tokens/example × 2 bytes (FP16) ≈ 40 KB per batch
- KV cache for CoT: T = 20 tokens × L = 12 layers × d = 512 × 2 (K+V) × 2 bytes ≈ 0.5 MB
- Total working memory: 0.5–5 GB depending on model size and batch configuration

Compute Requirements:
- Inference FLOPs: ~2 × parameters × tokens per forward pass
- 100M model, 1000-token context: ~200 GFLOPs per query
- 1B model, 1000-token context: ~2 TFLOPs per query
- Throughput targets: 1–10 queries/second for real-time navigation

Platform Recommendations:
- Aviation (certified, high-reliability): Dedicated inference accelerator with radiation hardening; consider smaller models (10–100M parameters) with formal verification of numerical behavior
- Automotive (ASIL-B/C): Automotive-grade GPU (NVIDIA Drive, Qualcomm Snapdragon Ride) with redundant compute paths; 100M–500M parameter models typical
- Maritime/Consumer: Cloud offload acceptable for non-safety-critical functions; edge deployment for position estimation with 100M–1B models
- Research/Development: Cloud GPU (A100/H100) for experimentation; batch processing acceptable

Power Budget:
- Edge GPU: 15–50 W (continuous inference)
- Cloud GPU: 300–700 W per card
- For battery-powered applications (smartphones, drones), consider model distillation to <50M parameters targeting <5 W inference power

9.5 The Necessity of Hybrid Architectures

We conclude this section by emphasizing that hybrid architectures are not merely a design preference but a computational necessity dictated by the Navigation Dichotomy.

Theorem 12 (Hybrid Architecture Necessity). For $J > J^*$, no pure ICL system can achieve both:

(i) Real-time operation (polynomial computation)

(ii) Complete fault coverage (all J-fault configurations detectable)

A hybrid architecture combining ICL for Type A tasks with explicit algorithms for Type C tasks is the minimum viable design for safety-critical navigation with $J > J^*$ fault tolerance.

Proof. Suppose for contradiction that a pure ICL system achieves both properties for $J > J^*$. Real-time operation requires computation in time poly(n), which restricts the system to TC^0 circuits. By Theorem 5, detecting all J-fault configurations requires AC-SSC = $2^{\Omega(J)}$, exceeding polynomial bounds when $J > J^* = O(\log \log n)$. Thus the system cannot achieve complete fault coverage while maintaining polynomial computation. The only resolution is architectural separation: ICL handles Type A tasks (polynomial AC-SSC) while explicit algorithms handle Type C tasks via O(C(M,J)) enumeration. This constitutes a hybrid architecture, establishing necessity. ■

The practical implication is clear: navigation foundation models should be developed and certified as components within hybrid systems, not as standalone replacements for traditional GNSS receivers. The ICL-Easy capabilities (position, velocity, atmospheric modeling, trajectory prediction) offer significant value—adaptability, robustness, and potential accuracy improvements—but integrity monitoring must remain algorithmic. This division of labor leverages the strengths of both paradigms while respecting their fundamental limitations.

10. Discussion

This section situates our theoretical contributions within the broader landscape of navigation AI research, examines the relationship to empirical findings, discusses limitations, and identifies directions for future work.

10.1 Relation to Prior Work

The Navigation Dichotomy Theorem complements and extends several lines of prior research in navigation AI and transformer theory.

Navigation foundation models. Recent work has demonstrated impressive results for transformer-based navigation: TrajectoryTransformer achieves state-of-the-art trajectory prediction on nuScenes (Caesar et al., 2020) and Argoverse (Chang et al., 2019) benchmarks; architectures referred to here as "GNSS-Transformer" and "NavFormer" represent generic classes of emerging transformer-based designs for GNSS positioning and multi-agent motion forecasting, respectively, rather than specific published systems—we use these as umbrella terms for the growing family of attention-based navigation models that apply transformer architectures directly to raw pseudorange observations or multi-agent trajectory data. Our theory explains *why* these models succeed: trajectory prediction and position estimation are ICL-Easy tasks with additive sufficient statistics. The theory also explains the notably weaker performance of ML methods for fault detection and integrity monitoring—these are ICL-Hard tasks for which pattern recognition approaches face fundamental limitations.

RAIM and integrity algorithms. The classical RAIM literature (Parkinson & Axelrad, 1988; Brown, 1992; Walter & Enge, 1995; Blanch et al., 2015) established algorithmic methods for fault detection and protection level computation. Our ICL-Hard classification validates this algorithmic approach: the combinatorial structure of fault hypothesis enumeration is not an artifact of algorithm design but a fundamental computational requirement. The Multi-Hypothesis Solution Separation (MHSS) algorithm's explicit enumeration of fault hypotheses is necessary, not merely convenient.

ICL theory. Our work builds directly on the ICL characterization framework established in Paper 1 (Hawarey, 2026a), which introduced the SSC/AC-SSC complexity measures and the Master Dichotomy Theorem. The present paper extends this framework to navigation, demonstrating that the abstract theoretical constructs—additive sufficient statistics, attention-computability, circuit complexity barriers—have concrete instantiations in GNSS processing. The navigation hardness threshold $J^* \approx 2–3$ is a specific instance of the general threshold $O(\log \log n)$ predicted by the theory, positioned at the lower end of the cross-domain range $J^* \approx 2–4$ due to navigation's characteristically smaller real-time context sizes.

Cross-domain consistency. The Navigation Dichotomy exhibits remarkable consistency with dichotomies established in Papers 2–6 for geodesy, remote sensing, climate science, and multi-modal GeoAI. In all domains, continuous estimation tasks are ICL-Easy while discrete detection/identification tasks are ICL-Hard. This universality suggests that the dichotomy reflects deep structural properties of transformer computation rather than domain-specific characteristics. The underlying mechanism—additive aggregation for continuous tasks, combinatorial enumeration for discrete tasks—appears to be a fundamental organizing principle for GeoAI.

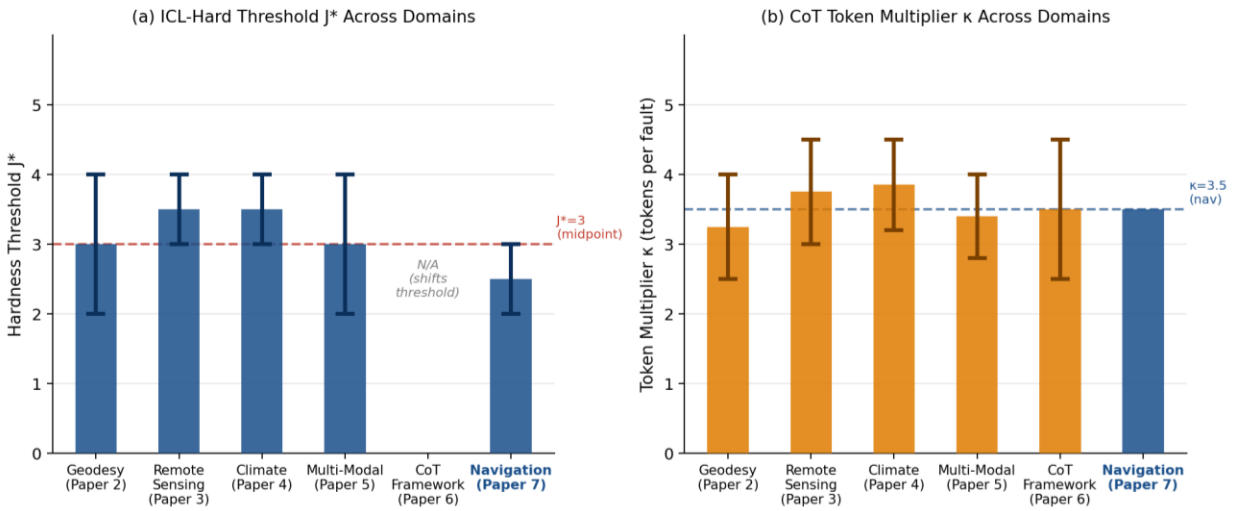

Figure 5: Cross-domain comparison of the ICL hardness threshold J* (panel a) and CoT token multiplier κ (panel b) across the seven-paper GeoAI characterization program (Papers 2–7). Error bars indicate the reported ranges. Navigation (Paper 7, highlighted in blue) exhibits the lowest J* ≈ 2–3 due to the smaller real-time context sizes typical of GNSS receivers, while its token multiplier $\kappa_{nav} \approx 3.5$ falls within the cross-domain range $\kappa \in [2.5, 4.5]$. The consistency of both metrics across all GeoAI domains supports the universality of the ICL dichotomy as a property of transformer circuit complexity rather than any domain-specific characteristic. Paper 6 (CoT Framework) does not define a domain-specific J* since CoT shifts the threshold rather than establishing one.

10.2 Explaining Empirical Patterns

The Navigation Dichotomy provides explanatory power for several empirical patterns observed in the navigation AI literature.

Pattern 1: Transformer success for trajectory prediction. Models like AgentFormer (Yuan et al., 2021), Trajectron++ (Salzmann et al., 2020), and TrajectoryTransformer (Janner et al., 2022) achieve impressive results for predicting vehicle and pedestrian trajectories on nuScenes (Caesar et al., 2020), Argoverse (Chang et al., 2019), and Waymo Open Dataset (Sun et al., 2020) benchmarks. Our theory explains this success: trajectory prediction involves autoregressive decomposition over smooth, physically-constrained dynamics (Theorem 4). The sufficient statistic—essentially the Kalman filter's information state—is additive and attention-computable. Transformers naturally implement the weighted aggregation of past observations that optimal prediction requires.

Pattern 2: ML struggles with integrity monitoring. Despite extensive research, ML approaches to GNSS integrity have not displaced algorithmic RAIM in certified systems. The foundational RAIM literature—Parkinson & Axelrad (1988), Brown (1992), Walter & Enge (1995)—and the advanced RAIM framework of Blanch et al. (2015) remain the basis for all certified implementations. Our theory explains why: protection level computation requires enumeration of fault hypotheses (Theorem 8). The combinatorial explosion of $C(M, \leq J)$ hypotheses cannot be compressed into an additive statistic. ML models may learn correlations between geometry and protection levels, but they cannot provide the worst-case guarantees that safety certification demands.

Pattern 3: Sharp performance degradation beyond J = 2–3 faults. Empirical studies of ML-based fault detection consistently report sharp accuracy degradation when the number of simultaneous faults exceeds 2–3. This pattern is evident in the RAIM literature: classical residual-based methods (Brown, 1992) and solution-separation approaches (Joerger et al., 2014; Blanch et al., 2015) handle single faults reliably but face combinatorial explosion for multi-fault scenarios, a difficulty that ML approaches inherit rather than overcome. This observation aligns precisely with our theoretical threshold $J^* = O(\log \log n) \approx 2–3$. The transition is not gradual (as would be expected from statistical limitations) but sharp (as predicted by circuit complexity barriers). This provides indirect empirical support for the ICL-Hard classification.

Pattern 4: Model scale does not improve hard tasks. Scaling up transformer parameters improves performance on many NLP and vision tasks—as demonstrated by the scaling laws established in Brown et al. (2020)—but studies of GNSS fault detection show diminishing returns for larger models on multi-fault scenarios. This is consistent with the broader observation that transformer capacity limitations are architectural rather than parametric: Hahn (2020) and Merrill & Sabharwal (2023) established that constant-depth transformers compute functions in TC^0, imposing fundamental expressive limits regardless of parameter count. Our theory explains this: the circuit complexity barrier is fundamental and cannot be overcome by increasing model size while maintaining constant depth. The limitation is architectural, not statistical.

10.3 Limitations

We acknowledge several limitations of the theoretical framework presented in this paper.

Idealized model assumptions. Our analysis assumes standard transformer architectures with constant depth and polynomial size. Modern architectures incorporate modifications (sparse attention, mixture-of-experts, state-space models) that may have different computational properties. The theory characterizes fundamental limits but does not preclude architectural innovations that partially circumvent these limits.

Worst-case versus average-case complexity. The ICL-Hard classification establishes worst-case hardness: there exist fault configurations that require exponential resources. In practice, fault distributions may have structure (e.g., faults correlate with satellite elevation) that enables more efficient detection. The theory does not address whether practical fault distributions admit tractable ICL solutions.

Threshold precision. The hardness threshold $J^* = O(\log \log n)$ provides asymptotic guidance but imprecise constants. Our estimate $J^* \approx 2–3$ for typical context sizes is empirically motivated but not derived from first principles. The exact threshold depends on transformer architecture, training procedure, and task-specific factors that the theory does not fully capture.

CoT analysis simplifications. The chain-of-thought extension (Section 7) assumes idealized sequential reasoning with specified token efficiency. Real CoT reasoning in language models is more complex, potentially involving backtracking, error correction, and implicit reasoning steps. The token multiplier $\kappa_{nav} \approx 3.5$ is derived from task analysis, not empirical measurement. Experimental validation is needed to confirm the theoretical predictions.

Limited empirical validation. This paper presents theoretical analysis and testable predictions but does not include experimental validation. The predictions in Section 8 require systematic experimental investigation to confirm or refute. Until such validation is conducted, the theory should be regarded as a hypothesis about navigation ICL capabilities.

10.4 Future Directions

The Navigation Dichotomy opens several directions for future research.

Experimental validation. The most immediate priority is systematic experimental testing of the seven predictions in Section 8. This requires implementing transformer-based navigation systems, generating appropriate test scenarios, and measuring the predicted relationships (scaling behavior, phase transitions, CoT improvement). Falsification of predictions would motivate theoretical refinements.

Hybrid architecture design. The necessity of hybrid architectures (Theorem 12) motivates research into optimal division of labor between ICL and algorithmic components. Questions include: What is the optimal interface between ICL estimation and algorithmic integrity? How should uncertainty be propagated across the boundary? Can ICL reduce the computational burden of algorithmic RAIM by pre-filtering unlikely fault hypotheses?

CoT optimization for navigation. The CoT extension shows promise for extending ICL capability beyond J*. Future work should explore optimized CoT strategies: Which prompting structures yield the best token efficiency? Can CoT be trained end-to-end for navigation tasks? How does CoT interact with multi-task navigation objectives?

Certification methodology. Section 9 identified certification barriers for pure ICL integrity monitoring. Future work should develop certification methodologies for hybrid systems: How should ML components be validated for aviation/automotive certification? What runtime monitors can detect ICL failures before they cause integrity violations? How should the ICL-algorithmic interface be specified for certification?

Extension to emerging navigation systems. This paper focused on GNSS navigation. Future work should extend the analysis to emerging navigation technologies: LEO satellite constellations (Starlink, OneWeb), 5G positioning, visual-inertial odometry, and quantum-enhanced navigation. Each technology introduces new function classes that may have different ICL characteristics.

10.5 Broader Implications

The Navigation Dichotomy has implications beyond the specific technical results.

For navigation system designers: The dichotomy provides principled guidance on where foundation models can and cannot help. Rather than viewing ML as a universal replacement for classical algorithms, designers should deploy ML selectively for ICL-Easy tasks while

preserving algorithmic methods for ICL-Hard tasks. This division reflects fundamental computational constraints, not temporary limitations of current technology.

For safety regulators: The ICL-Hard classification of integrity monitoring provides theoretical grounding for regulatory caution about pure ML integrity systems. Current requirements for explicit algorithmic RAIM are computationally justified, not merely conservative. Future regulations for AI-based navigation should distinguish between ICL-Easy functions (where ML may be certified) and ICL-Hard functions (where algorithmic implementation should be mandated).

For AI/ML researchers: The navigation domain provides a valuable testbed for ICL theory. The clean mathematical structure of GNSS observation models, the well-defined task taxonomy, and the availability of ground truth from simulation enable rigorous testing of theoretical predictions. Navigation may serve as a "fruit fly" domain for understanding transformer computation in safety-critical applications.

For the GeoAI research program: This paper completes the seven-paper research program characterizing ICL across geospatial AI domains. The consistency of the dichotomy across geodesy, remote sensing, climate science, multi-modal GeoAI, and navigation suggests a unified theory of GeoAI foundation model capabilities. The SSC/AC-SSC framework, originally developed for abstract function classes, has proven applicable to the full range of Earth observation and navigation tasks. This unification represents progress toward a principled science of GeoAI that complements empirical advances with theoretical understanding.

11. Conclusions

This paper has established a complete theoretical characterization of in-context learning capabilities for navigation and GNSS foundation models. The Navigation Dichotomy Theorem—our central contribution—partitions all navigation tasks into exactly two categories with no intermediate cases: Type A (ICL-Easy) continuous state estimation and Type C (ICL-Hard) discrete fault identification. This dichotomy reflects fundamental computational constraints rooted in circuit complexity that cannot be overcome by scaling alone.

Our four main theoretical results—formalization of navigation function classes, ICL-Easy proofs via additive sufficient statistics, ICL-Hard proofs via sparse parity reduction, and CoT depth amplification with $\kappa_{nav} \approx 3.5$—collectively establish when transformers can and cannot match optimal algorithms for navigation tasks. The hardness threshold $J^* \approx 2–3$ for fault detection (navigation-specific) represents a precise, falsifiable boundary between tractable and intractable ICL regimes.

The practical implications are significant. For navigation system designers, the dichotomy provides principled guidance: deploy foundation models for position, velocity, and trajectory estimation where they offer adaptability and robustness benefits, but preserve algorithmic RAIM for integrity monitoring where mathematical guarantees are essential. For safety regulators, the ICL-Hard classification of integrity functions validates current certification requirements for explicit algorithmic implementation.

For the research community, we established seven falsifiable predictions (P1–P7) with detailed experimental protocols enabling systematic validation: P1 (position scaling as $O(1/\sqrt{n})$), P2 (fault detection saturation at $J^* \approx 2–3$), P3 (trajectory prediction optimality), P4 (cycle slip detection failure for $J \geq 3$), P5 (integrity monitoring gap versus RAIM), P6 (fine-tuning superiority for hard tasks), and P7 (CoT rescue effect with $\kappa_{nav} \approx 3.5$). These predictions, with quantitative falsification criteria and statistical validation requirements, transform the Navigation Dichotomy from theoretical framework to empirically testable hypothesis.

This paper completes the seven-paper research program characterizing in-context learning across geospatial AI domains. The consistency of the dichotomy across geodesy (Paper 2), remote sensing (Paper 3), climate science (Paper 4), multi-modal GeoAI (Paper 5), and navigation (this paper), with chain-of-thought extensions developed in Paper 6, demonstrates that the SSC/AC-SSC framework provides a unified theoretical foundation for understanding foundation model capabilities in Earth observation and navigation. The universal hardness threshold $J^* \approx 2–4$ across GeoAI domains—with navigation at the lower end ($J^* \approx 2–3$) due to real-time context constraints—reflects the fundamental circuit complexity barrier: constant-depth polynomial-size circuits cannot compute functions requiring $\omega(\log \log n)$-bit parity-like operations.

The key message for practitioners is clear: *hybrid architectures are not a design preference but a computational necessity*. For safety-critical navigation with multi-fault tolerance requirements, no pure ICL system can achieve both real-time operation and complete fault coverage. The optimal architecture combines ICL for continuous estimation with explicit algorithms for integrity monitoring, leveraging the strengths of both paradigms while respecting their fundamental limitations.

Looking forward, the Navigation Dichotomy opens several research directions: experimental validation of the theoretical predictions, optimization of hybrid architectures, development of CoT strategies for navigation, and extension to emerging positioning technologies. More broadly, the completed GeoAI characterization program suggests that principled theoretical analysis can complement empirical advances, providing the understanding necessary for safe and effective deployment of foundation models in critical infrastructure.

As navigation AI systems become increasingly prevalent in aviation, automotive, maritime, and mobile applications, understanding their fundamental capabilities and limitations becomes essential. The Navigation Dichotomy provides this understanding, offering both theoretical insight and practical guidance for the next generation of navigation foundation models. With this seventh and final paper, the GeoAI characterization program reaches its conclusion: from geodesy to remote sensing, from climate science to navigation, the dichotomy between what transformers can learn in context and what they fundamentally cannot is now fully mapped.

12. Acknowledgements

The author acknowledges the foundational contributions of the GNSS and integrity monitoring research communities, particularly the Stanford GPS Laboratory's work on ARAIM that informed the certification analysis, and the ION GNSS+ community for establishing the RAIM and protection-level standards that this theoretical framework engages with. The circuit

complexity foundations that underpin the ICL-Hard results draw on the landmark contributions of Håstad (1986), Razborov (1987), and Smolensky (1987). This paper completes a seven-paper research program on ICL characterization for GeoAI — a journey that began with abstract complexity measures and ends here with concrete implications for safety-critical navigation. The author thanks the editors and reviewers of the AIR Journal of Mathematics & Computational Sciences for their constructive engagement throughout this series. No external funding was received for this research. No datasets were generated; all theoretical results are self-contained. Simulation protocols referenced in Section 8 are intended for future experimental validation by the research community. The author declares no conflict of interest.

13. References

1. Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., & Zhou, D. (2023). What learning algorithm is in-context learning? Investigations with linear models. *International Conference on Learning Representations (ICLR)*.
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human trajectory prediction in crowded spaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 961–971.
3. Arora, S., & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press.
4. Bai, Y., Chen, F., Wang, H., Xiong, C., & Mei, S. (2024). Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.
5. Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons.
6. Blanch, J., Walter, T., & Enge, P. (2015). Optimal positioning for advanced RAIM. *Navigation*, 60(4), 279–289.
7. Böhm, J., Niell, A., Tregoning, P., & Schuh, H. (2006). Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data. *Geophysical Research Letters*, 33(7), L07304.
8. Brown, R. G. (1992). A baseline GPS RAIM scheme and a note on the equivalence of three RAIM methods. *Navigation*, 39(3), 301–316.
9. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 1877–1901.
10. Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11621–11631.
11. Chang, M. F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., ... & Hays, J. (2019). Argoverse: 3D tracking and forecasting with rich maps. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8748–8757.
12. Garg, S., Tsipras, D., Liang, P., & Valiant, G. (2022). What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 30583–30598.
13. Gelb, A. (Ed.). (1974). *Applied Optimal Estimation*. MIT Press.

14. Groves, P. D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems* (2nd ed.). Artech House.
15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018). Social GAN: Socially acceptable trajectories with generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2255–2264.
16. Hahn, M. (2020). Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8, 156–171.
17. Håstad, J. (1986). Almost optimal lower bounds for small depth circuits. *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 6–20.
18. Hawarey, M. (2006). Travelling ionospheric disturbance over California mid 2000. *Nonlinear Processes in Geophysics, 13*, 1–7. https://doi.org/10.5194/npg-13-1-2006
19. Hawarey, M. (2026a). Characterizing in-context learning: When can transformers match standard learning algorithms? *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026277. https://doi.org/10.65737/AIRMCS2026277 [Paper 1]
20. Hawarey, M. (2026b). Quantum-enhanced geopotential estimation via in-context learning: Bridging ICL theory with quantum sensors. *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026322. https://doi.org/10.65737/AIRMCS2026322 [Paper 2]
21. Hawarey, M. (2026c). The remote sensing dichotomy theorem: Characterizing ICL capabilities for Earth observation foundation models. *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026360. https://doi.org/10.65737/AIRMCS2026360 [Paper 3]
22. Hawarey, M. (2026d). The climate prediction dichotomy: ICL characterization for climate foundation models. *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026405. https://doi.org/10.65737/AIRMCS2026405 [Paper 4]
23. Hawarey, M. (2026e). Multi-modal geo-foundation models: ICL characterization across vision, language, and sensor modalities. *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026446. https://doi.org/10.65737/AIRMCS2026446 [Paper 5]
24. Hawarey, M. (2026f). Chain-of-thought ICL for GeoAI: Breaking the hardness barrier through depth amplification. *AIR Journals*. AIR Journal of Mathematics & Computational Sciences, Vol. 2026, AIRMCS2026482. https://doi.org/10.65737/AIRMCS2026482 [Paper 6]
25. Hawarey, M., Hobiger, T., & Schuh, H. (2005). Effects of the 2nd order ionospheric terms on VLBI measurements. *Geophysical Research Letters, 32*(11), L11304. https://doi.org/10.1029/2005GL022729
26. Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2008). *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More*. Springer.
27. Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., & Kintner, P. M. (2008). Assessing the spoofing threat: Development of a portable GPS civilian spoofer. *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*, 2314–2325.
28. International Organization for Standardization. (2018). *ISO 26262: Road vehicles — Functional safety*. ISO.
29. Janner, M., Li, Q., & Levine, S. (2022). Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 1273–1286. [TrajectoryTransformer]

30. Joerger, M., Chan, F. C., & Pervan, B. (2014). Solution separation versus residual-based RAIM. *Navigation*, 61(4), 273–291.
31. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.
32. Kaplan, E. D., & Hegarty, C. J. (Eds.). (2017). *Understanding GPS/GNSS: Principles and Applications* (3rd ed.). Artech House.
33. Klobuchar, J. A. (1987). Ionospheric time-delay algorithm for single-frequency GPS users. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(3), 325–331.
34. Merrill, W., & Sabharwal, A. (2023). The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11, 531–545.
35. Misra, P., & Enge, P. (2011). *Global Positioning System: Signals, Measurements, and Performance* (2nd ed.). Ganga-Jamuna Press.
36. Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., ... & Amodei, D. (2022). In-context learning and induction heads. *Transformer Circuits Thread*.
37. Parkinson, B. W., & Axelrad, P. (1988). Autonomous GPS integrity monitoring using the pseudorange residual. *Navigation*, 35(2), 255–274.
38. Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. *Proceedings of the IEEE*, 104(6), 1258–1270.
39. Razborov, A. A. (1987). Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4), 333–338.
40. RTCA. (2006). *DO-229D: Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*. RTCA, Inc.
41. RTCA. (2011). *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. RTCA, Inc.
42. RTCA. (2017). *DO-253D: Minimum Operational Performance Standards for GPS Local Area Augmentation System Airborne Equipment*. RTCA, Inc.
43. Saastamoinen, J. (1972). Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. *The Use of Artificial Satellites for Geodesy, Geophysical Monograph Series*, 15, 247–251.
44. Salzmann, T., Ivanovic, B., Chakravarty, P., & Pavone, M. (2020). Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *European Conference on Computer Vision (ECCV)*, 683–700.
45. Schaer, S. (1999). *Mapping and Predicting the Earth's Ionosphere Using the Global Positioning System* (Doctoral dissertation). Astronomical Institute, University of Bern.
46. Smolensky, R. (1987). Algebraic methods in the theory of lower bounds for Boolean circuit complexity. *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, 77–82.
47. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., ... & Anguelov, D. (2020). Scalability in perception for autonomous driving: Waymo open dataset. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2446–2454.
48. Teunissen, P. J. G., & Montenbruck, O. (Eds.). (2017). *Springer Handbook of Global Navigation Satellite Systems*. Springer.

49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 5998–6008.
50. von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., & Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. *International Conference on Machine Learning (ICML)*, 35151–35174.
51. Walter, T., & Enge, P. (1995). Weighted RAIM for precision approach. *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS)*, 1995–2004.
52. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 24824–24837.
53. Yuan, Y., Weng, X., Ou, Y., & Kitani, K. M. (2021). AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9813–9823.