Characterizing the Representational Capacity of Neural Processes

Robin Young[1,†]

[1]University of Cambridge
[†]Correspondence to robin.young@cl.cam.ac.uk

Abstract

What functions can Neural Processes represent? We analyze the representational capacity of popular NP architectures: Conditional Neural Processes (CNPs), Attentive Neural Processes (ANPs), Transformer Neural Processes (TNPs), and their latent variants. We prove these architectures form a strict hierarchy. CNP-representable functions are exactly those depending on finitely many expected features of the context distribution. ANPs strictly generalize CNPs via query-dependent reweighting, enabling kernel smoothers. ConvCNPs and ANPs are incomparable; each contains functions outside the other, separated by stationarity versus translation equivariance. TNPs with L self-attention layers capture L-hop context interactions. For latent NPs, we show finite-dimensional latents provide coherent sampling but do not circumvent encoder limitations; matching GP posterior distributions requires latent dimension scaling with context size. These results provide a theoretical foundation for architecture selection based on task structure.

1. Introduction

Neural Processes (Garnelo et al., 2018a,b) have emerged as a family of models for meta-learning and few-shot prediction. By learning to map context sets to predictive distributions, NPs combine the computational efficiency of neural networks with the uncertainty quantification traditionally associated with Gaussian Processes (GPs).

Since their introduction, numerous architectural variants have been proposed including Conditional Neural Processes (CNPs), Attentive Neural Processes (ANPs) (Kim et al., 2019), Convolutional CNPs (Gordon et al., 2020), and Transformer Neural Processes (TNPs) (Nguyen and Grover, 2022). Many have observed that these architectures exhibit different capabilities. ANPs outperform CNPs on tasks requiring local adaptation, ConvCNPs excel on spatially structured stationary tasks, while TNPs excel at tasks requiring global coherence, but the theoretical foundations explaining these differences have been lacking.

In this paper, we answer a natural question: *What functions can neural processes represent?*

We provide a characterization of CNP-representable functions as those depending on finitely many expected features of the empirical context distribution. We prove that ANPs strictly generalize CNPs by enabling query-dependent reweighting, with an explicit construction showing that kernel smoothers are ANP-representable but lie outside the CNP function class. We show that ConvCNPs and ANPs are incomparable in the hierarchy with the separation governed by stationarity versus translation equivariance. For TNPs, we establish that L layers of self-attention capture L-hop context interactions, and prove matching upper and lower bounds: $\Theta(\sqrt{\kappa}\log(1/\varepsilon))$ layers are necessary

and sufficient to ε-approximate GP posteriors, where κ is the kernel matrix condition number. We extend our analysis to latent NPs, showing that finite-dimensional latent variables provide coherent function sampling but do not expand representational capacity beyond what the encoder permits.

2. Related Work

Neural Processes were introduced by Garnelo et al. (2018a) as a computationally efficient alternative to Gaussian Processes for meta-learning, with the latent variable extension following in Garnelo et al. (2018b). Subsequent work has developed numerous architectural variants. Attentive Neural Processes (Kim et al., 2019) replaced mean aggregation with cross-attention, allowing the context representation to depend on the target location. Convolutional Conditional Neural Processes (Gordon et al., 2020) introduced translation equivariance through convolutional aggregation, achieving strong performance on spatially structured tasks. Transformer Neural Processes (Nguyen and Grover, 2022) applied self-attention to the context set before cross-attention, enabling context points to exchange information. Practitioners have observed that these architectures exhibit qualitatively different capabilities, but the theoretical foundations explaining these differences have been absent. We provide the first rigorous characterization of the representational capacity of each architecture class, proving they form a strict hierarchy.

The representation of functions over sets has been studied extensively in the deep learning literature. Zaheer et al. (2017) established that permutation invariant functions can be represented by DeepSets architectures with sum or mean aggregation, while follow up work by Wagstaff et al. (2019) proved limitations that no continuous function with fixed-dimension aggregation can be injective on sets of unbounded size. Our work extends these results to the predictive setting where outputs depend on both the context set and a target location. The distinction is that Neural Processes must produce query-dependent predictions, not just set-level summaries. We show that query-dependent reweighting strictly expands representational capacity even when the aggregated dimension is held fixed.

Gaussian Processes are standard for uncertainty quantification in regression, but exact inference scales cubically in the number of observations. This has motivated extensive work on sparse approximations, including inducing point methods (Snelson and Ghahramani, 2005; Titsias, 2009), random Fourier features (Rahimi and Recht, 2007), and structured kernel interpolation (Wilson and Nickisch, 2015). Neural Processes take a different approach as they learn to amortize GP-like inference through an encoder-decoder architecture trained across tasks. Our results characterize when this amortization is possible. We prove that CNPs and ANPs cannot represent GP posteriors regardless of encoder capacity (Theorems 6 and 12), while TNPs can approximate GP posteriors with depth scaling as $\Theta(\sqrt{\kappa}\log(1/\varepsilon))$ where κ is the kernel matrix condition number (Theorem 15, Proposition 14). This provides a precise sense in which TNPs succeed where simpler architectures fail.

The representational capacity of transformer architectures has received significant recent attention. Yun et al. (2020) proved that transformers are universal approximators of sequence-to-sequence functions, while Pérez et al. (2021) established Turing completeness under suitable assumptions. Our depth lower bound (Theorem 15) contributes to this literature by proving task-specific depth requirements. Approximating GP posteriors requires $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ layers regardless of width. The key insight is that linearization at $y_C = 0$ reduces the Jacobian to a polynomial in the attention matrix, enabling the application of classical approximation barriers. This technique may be applicable to other regression tasks requiring matrix inversion.

Our depth lower bound relies on classical results from approximation theory, particularly polynomial approximation of rational functions. The Chebyshev barrier for approximating $1/\mu$ on an interval dates to the foundational work of Chebyshev and Markov; see Trefethen (2019) for a modern treatment. The connection between iterative methods and polynomial approximation is well-established in numerical linear algebra, where Chebyshev iteration achieves optimal convergence rates for linear

systems (Golub and Van Loan, 2013). Our contribution is recognizing that transformer self-attention layers implement polynomial iterations, and that the same barriers governing classical iterative methods govern neural network depth requirements. The matching upper bound (Proposition 14) shows that the Chebyshev polynomial construction can be realized by appropriate choice of attention weights.

Theoretical analysis of meta-learning has developed along several axes. Generalization bounds for learning-to-learn were established by Baxter (2000), with subsequent refinements using PAC-Bayes (Pentina and Lampert, 2014; Rothfuss et al., 2021) and information-theoretic (Jose and Simeone, 2021) techniques. These results bound how many tasks are needed for a meta-learner to generalize to new tasks. Our work is complementary. We characterize what can be represented, not how fast it can be learned. The expressiveness hierarchy we establish has implications for meta-learning theory. A CNP cannot benefit from additional tasks if the target predictor lies outside its function class, but a full sample complexity analysis for Neural Processes remains open.

3. Preliminaries

Let $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ be the input space and $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$ the output space. A context set is a finite collection $C = \{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$. The context space $\mathcal{C} = \bigcup_{n=1}^\infty (\mathcal{X} \times \mathcal{Y})^n / S_n$ is the set of all finite multisets, with $\mathcal{C}_{\leq N}$ the restriction to size at most N. A *predictive map* $F : \mathcal{C} \times \mathcal{X} \to \Theta$ returns distribution parameters $\theta \in \Theta$ given a context set C and target location x_t.

Conditional Neural Process (CNP). A CNP computes:

$$h_i = h_\phi(x_i, y_i) \in \mathbb{R}^d \qquad \text{(encode each context point)} \tag{1}$$

$$r_C = \frac{1}{n} \sum_{i=1}^n h_i \in \mathbb{R}^d \qquad \text{(aggregate via mean)} \tag{2}$$

$$\theta = g_\psi(x_t, r_C) \in \Theta \qquad \text{(decode at target)} \tag{3}$$

where $h_\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is the encoder and $g_\psi : \mathcal{X} \times \mathbb{R}^d \to \Theta$ is the decoder.

Attentive Neural Process (ANP). An ANP uses cross-attention from target to context:

$$\alpha_i(x_t; C) = \frac{\exp(q(x_t)^\top k(x_i, y_i)/\tau)}{\sum_{j=1}^n \exp(q(x_t)^\top k(x_j, y_j)/\tau)} \tag{4}$$

$$r_C(x_t) = \sum_{i=1}^n \alpha_i(x_t; C)\, v(x_i, y_i) \tag{5}$$

$$\theta = g_\psi(x_t, r_C(x_t)) \tag{6}$$

where $q : \mathcal{X} \to \mathbb{R}^{d_k}$ is the query network, $k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_k}$ is the key network, and $v : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is the value network.

Transformer Neural Process (TNP). A TNP applies self-attention to the context before cross-attention:

$$\beta_{ij}^{(\ell)} = \frac{\exp(q_s(\tilde{h}_i^{(\ell-1)})^\top k_s(\tilde{h}_j^{(\ell-1)})/\tau)}{\sum_{m=1}^n \exp(q_s(\tilde{h}_i^{(\ell-1)})^\top k_s(\tilde{h}_m^{(\ell-1)})/\tau)} \tag{7}$$

$$\tilde{h}_i^{(\ell)} = \sum_{j=1}^n \beta_{ij}^{(\ell)}\, W_v \tilde{h}_j^{(\ell-1)} \tag{8}$$

for layers $\ell = 1, \ldots, L$, with $\tilde{h}_i^{(0)} = h(x_i, y_i)$. The final representation is then used in cross-attention as in ANP.

Convolutional Neural Process (ConvCNP). A ConvCNP replaces finite-dimensional aggregation with functional convolutional channels. A non-negative filter $w : \mathbb{R}^{d_x} \to \mathbb{R}_+$ and pointwise encoder $h : \mathcal{Y} \to \mathbb{R}^d$ produce:

$$\rho_C(x) = \sum_{i=1}^{n} w(x - x_i) \qquad \text{(density channel)} \tag{9}$$

$$s_C(x) = \sum_{i=1}^{n} w(x - x_i)\, h(y_i) \qquad \text{(signal channel)} \tag{10}$$

$$\tilde{r}_C = \Phi(s_C, \rho_C) \qquad \text{(CNN processing)} \tag{11}$$

$$\theta = g(\tilde{r}_C(x_t), x_t) \qquad \text{(readout at target)} \tag{12}$$

where Φ is a multi-layer CNN. A pure ConvCNP omits the CNN: $\theta = g(s_C(x_t), \rho_C(x_t), x_t)$. We distinguish between the pure convolutional aggregation and the subsequent CNN processing, as these contribute qualitatively different representational capabilities.

The core question of representational capacity is basically: what information about the context can reach the prediction? For CNPs, only the mean-aggregated encoding and context points are processed independently. For ANPs, the query can reweight context points, but the weights factorize but context points still don't interact. ConvCNPs lift the finite-dimensional bottleneck via functional representations, and the CNN enables context–context coupling, but only through translation-equivariant operations. For TNPs, self-attention lets context points exchange information before the prediction is made. This progression from no interaction, to query-mediated reweighting, to full context-context coupling is what drives the hierarchy we establish.

4. Conditional Neural Processes

Definition 1 (h-Equivalence). *For a fixed encoder $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, two context sets C, C' are h-equivalent, written $C \sim_h C'$, if $\frac{1}{|C|}\sum_{(x,y)\in C} h(x,y) = \frac{1}{|C'|}\sum_{(x,y)\in C'} h(x,y)$.*

Proposition 2 (Indistinguishability). *A CNP with encoder h cannot distinguish h-equivalent context sets: if $C \sim_h C'$, then $\mathrm{CNP}(C, x_t) = \mathrm{CNP}(C', x_t)$ for all $x_t \in \mathcal{X}$.*

Theorem 3 (CNP Characterization). *The class of d-representable predictive maps is exactly the moment statistics class $\mathcal{F}^{(d)}_{\text{moment}} = \{F : F(C, x_t) = \phi(\frac{1}{n}\sum_i \psi(x_i, y_i), x_t)\}$ for continuous $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, $\phi : \mathbb{R}^d \times \mathcal{X} \to \Theta$.*

Proposition 4 (Existence of Collisions). *For any encoder $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ and any $n > d/(d_x + d_y)$, there exist distinct context sets $C \neq C'$ with $|C| = |C'| = n$ such that $C \sim_h C'$.*

Example 5 ($n = 2$ Collision). *Consider $d_x = d_y = 1$ and encoder $h(x, y) = (x, y) \in \mathbb{R}^2$. The contexts $C = \{(0, 1), (2, 1)\}$ and $C' = \{(0.5, 0.5), (1.5, 1.5)\}$ satisfy $\bar{h}_C = \bar{h}_{C'} = (1, 1)$, yet for an RBF kernel, $k(0, 2) \neq k(0.5, 1.5)$, so the GP posterior means differ. This two-point collision reappears in the ANP analysis (Theorem 12), where we show that even query-dependent reweighting cannot resolve it.*

Theorem 6 (GP Posterior is Not Finitely Representable). *A predictive map F is (d, ε)-representable on $\mathcal{C}_{\le n}$ if there exist continuous $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ and $g : \mathbb{R}^d \times \mathcal{X} \to \Theta$ such that $\sup_{C \in \mathcal{C}_{\le n}, x_t \in \mathcal{X}} \|F(C, x_t) - g(\frac{1}{|C|}\sum_{(x,y)\in C} h(x, y), x_t)\| \le \varepsilon$. Exact representability corresponds to $\varepsilon = 0$.*

For any d and any $n > d$, the GP posterior mean $\mu(x_t|C) = k(x_t, X_C)\mathbf{K}(X_C, X_C)^{-1} y_C$ is not $(d, 0)$-representable on $\mathcal{C}_{\le n}$ for generic positive definite kernels k.

Theorem 7 (CNP Approximation Lower Bound). *Let ν denote a distribution over target locations $x_t \in \mathcal{X}$, context locations $x_1, \ldots, x_n \in \mathcal{X}$ be fixed and let $y_C \sim \mathcal{N}(0, \mathbf{K})$. For the GP*

posterior mean, any CNP with representation dimension $d < n$ satisfies:

$$\inf_{\text{CNP}} \frac{\mathbb{E}_{y_C} \int |\mu(x_t|C) - \widehat{\mu}(x_t|C)|^2 \, d\nu(x_t)}{\mathbb{E}_{y_C} \int |\mu(x_t|C)|^2 \, d\nu(x_t)} \geq 1 - \frac{d}{n}$$

for any target distribution ν such that the whitened kernel vectors $\{\mathbf{K}^{-1/2} k(x_t, X_C)\}_{x_t \sim \nu}$ have isotropic second moment.

Example 8 (Linear Regression). *Even ordinary least squares illustrates the bottleneck. With k-dimensional features $\psi : \mathcal{X} \to \mathbb{R}^k$, the OLS predictor $F(C, x_t) = \langle (\sum_i \psi(x_i)\psi(x_i)^\top)^{-1} \sum_i y_i \psi(x_i), \psi(x_t) \rangle$ requires encoding both the Gram matrix $\sum_i \psi(x_i)\psi(x_i)^\top$ ($k(k+1)/2$ parameters) and the moment vector $\sum_i y_i \psi(x_i)$ (k parameters), giving $d = k(k+3)/2$. Since distinct Gram matrices generically yield distinct predictors, $d = \Omega(k^2)$ is also necessary.*

The bound in Theorem 7 is sharp and achieved by the PCA encoder $\bar{h}_C = A^* y_C$ that projects onto the top d principal components of the whitened weight covariance. For $d \ll n$, most GP posterior structure is lost regardless of encoder sophistication. The isotropic second moment condition holds whenever context and target locations are drawn i.i.d. from the same distribution and the kernel is stationary (e.g. RBF); for non-isotropic targets the bound generalizes to $\sum_{i=d+1}^n \lambda_i / \sum_{i=1}^n \lambda_i$ where λ_i are the eigenvalues of $\mathbb{E}_{x_t \sim \nu}[\tilde{w}_{x_t} \tilde{w}_{x_t}^\top]$. Full proofs appear in Appendix A.

5. Attentive Neural Processes

The key difference in ANPs is that the representation $r_C(x_t)$ depends on the query location x_t via attention weights $\alpha_i(x_t; C) \propto \exp(q(x_t)^\top k(x_i, y_i)/\tau)$. Unlike CNP-equivalence, ANP-equivalence is query-dependent: context sets equivalent at one query may be distinguishable at another.

Theorem 9 (ANPs Represent Kernel Smoothers). *For any continuous kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, the kernel smoother $F(C, x_t) = \sum_i K(x_t, x_i) y_i / \sum_i K(x_t, x_i)$ is ANP-representable to arbitrary precision with value dimension $d = d_y + 1$.*

Proof sketch. Set $v(x, y) = (y, 1)$, $k(x, y) = \psi(x)$, $q(x_t) = \phi(x_t)$, and choose ϕ, ψ so that $q(x_t)^\top k(x_i, y_i) \approx \log K(x_t, x_i)$. Then the attention weights recover the normalized kernel weights, and a linear decoder extracts the kernel smoother. The full construction and universal approximation argument appear in Appendix B. □

Theorem 10 (ANP Characterization). *A predictive map F is ANP-representable if and only if $F(C, x_t) = G(\mathbb{E}_{P_C^{x_t}}[v(x, y)], x_t)$ where $P_C^{x_t}(x_i, y_i) \propto \exp(s(x_t, x_i, y_i))$ for some continuous score function s.*

Corollary 11 (Strict Separation). $\mathcal{F}_{\text{CNP}}^{(d)} \subsetneq \mathcal{F}_{\text{ANP}}^{(d)}$ *for all d.*

Theorem 12 (GP Posterior Requires Context-Context Coupling). *For generic positive definite kernels, the GP posterior mean is not ANP-representable.*

Proof sketch. The GP posterior weight $w_i = [k(x_t, X_C)\mathbf{K}^{-1}]_i$ depends on the full Gram matrix, coupling all context points. ANP attention weights factor as $\alpha_i \propto f(x_t, x_i, y_i)$, depending only on query-point pairs independently. The simplest counterexample has $n = 2$: the GP weight on y_1 is

$$w_1 = \frac{k(x_t, x_1)k(x_2, x_2) - k(x_t, x_2)k(x_1, x_2)}{k(x_1, x_1)k(x_2, x_2) - k(x_1, x_2)^2},$$

which depends on the inter-context kernel value $k(x_1, x_2)$. No ANP attention weight $\alpha_1 \propto \exp(q(x_t)^\top k(x_1, y_1))$ can capture this coupling. See Appendix B for the general argument. □

6. Transformer Neural Processes

6.1. Polynomial Computation via Self-Attention

We analyze TNPs under two structural assumptions that enable clean characterization: position-based self-attention (attention weights depend only on input positions, not representations) and residual connections. These are relaxed in Section 6.3. Full definitions appear in Appendix C.

Theorem 13 (Polynomial Representation). *Under position-based attention with residual connections, after L self-attention layers with scalar value weights $W_v^{(\ell)} = \alpha_\ell \mathbf{I}$, the representation matrix satisfies $H^{(L)} = p_L(\tilde{\mathbf{K}})H^{(0)}$ where $p_L(\tilde{\mathbf{K}}) = \prod_{\ell=1}^{L}(\mathbf{I}+\alpha_\ell\tilde{\mathbf{K}})$ is a degree-L polynomial in the attention matrix $\tilde{\mathbf{K}}$.*

Proof. By induction: $H^{(L)} = (\mathbf{I} + \alpha_L\tilde{\mathbf{K}})H^{(L-1)} = \prod_{\ell=1}^{L}(\mathbf{I} + \alpha_\ell\tilde{\mathbf{K}})H^{(0)}$. □

6.2. Approximating GP Posteriors

The GP posterior involves $\mathbf{K}^{-1}$. Via the Neumann series $\mathbf{K}^{-1} = \frac{1}{\lambda_{\max}}\sum_{m=0}^{\infty}(\mathbf{I} - \mathbf{K}/\lambda_{\max})^m$, this can be approximated by matrix polynomials. The truncation error is $O(\rho^L/\lambda_{\min})$ where $\rho = 1-1/\kappa$, requiring $L = O(\kappa\log(1/\varepsilon))$ layers. The Chebyshev construction improves this to $O(\sqrt{\kappa}\log(1/\varepsilon))$:

Proposition 14 (Chebyshev Upper Bound). *There exist scalar value weights $\alpha_1, \ldots, \alpha_L$ such that $\|\prod_{\ell=1}^{L}(\mathbf{I} + \alpha_\ell\mathbf{K}) - \mathbf{K}^{-1}\| \le \frac{2}{\lambda_{\min}}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^L$.*

The weights are the optimal Chebyshev iteration parameters $\alpha_\ell = -2/(\lambda_{\max} + \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\cos\theta_\ell)$ with $\theta_\ell = (2\ell-1)\pi/(2L)$. See Appendix C.3 for the full proof and idealized TNP approximation construction.

6.3. Lower Bounds for Representation-Based Attention

The upper bound assumed position-based attention. We now show the $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ depth requirement holds even for fully adaptive representation-based attention.

The key insight is that the GP posterior mean $\mu(x_t|C) = k(x_t, X_C)\mathbf{K}^{-1}y_C$ is linear in y_C. Linearizing the TNP at $y_C = 0$ neutralizes adaptive routing: the Jacobian $\partial F/\partial y_C|_{y_C=0}$ becomes a polynomial in the base attention matrix (evaluated at $y_C = 0$), regardless of how sophisticated the attention mechanism.

Theorem 15 (TNP Depth Lower Bound). *For any $\kappa > 1$, there exists a family of context configurations with kernel matrices satisfying $\kappa(\mathbf{K}) = \kappa$ such that any L-layer TNP with representation-based self-attention achieving ε-approximation of $\mathbf{K}^{-1}y_C$ must satisfy:*

$$L \ge \frac{\sqrt{\kappa}}{4}\cdot\log\left(\frac{c}{\varepsilon}\right)$$

for a universal constant $c > 0$.

Proof sketch. The argument proceeds in four stages:

(1) Linearization. By Taylor expansion (Lemma 37 in Appendix D), ε-approximation of the linear target implies the Jacobian $M(X) = \partial F/\partial y_C|_{y_C=0}$ satisfies $\|M(X) - \mathbf{K}^{-1}\| \le O(\varepsilon)$.

(2) Polynomial structure. Differentiating through L self-attention layers shows $M(X)$ is a matrix polynomial of degree at most $2L$ in the attention matrix $\tilde{\mathbf{K}}|_{y_C=0}$ (Appendix D, Corollary 42).

(3) Univariate reduction. We construct a family of kernel matrices $\{\mathbf{K}_t\}$ (Lemma 45) where all eigenvalues except the minimum are equal to 1, with condition number κ. On this family, the quadratic form $v_1^\top M(\mathbf{K}_t)v_1$ reduces to a univariate polynomial in the minimum eigenvalue $\mu_1(t)$.

(4) Chebyshev barrier. The target $v_1^\top \mathbf{K}_t^{-1} v_1 = 1/\mu_1(t)$ must be approximated by a degree-$2L$ polynomial on $[1/\kappa, 1]$. The classical Chebyshev lower bound gives error at least $\Omega(\rho^{2L})$ where $\rho = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$. Solving for L yields the stated bound. □

The full proof, including the Jacobian evolution through representation-based attention layers, the eigenvalue-controlled kernel family construction, and spectral analysis, appears in Appendix D.

Corollary 16 (Tight Characterization). *For well-conditioned context geometries, $L = \Theta(\sqrt{\kappa}\log(1/\varepsilon))$ self-attention layers are both necessary and sufficient for ε-accurate GP posterior approximation.*

Remark 17 (Multi-Head Attention). *The analysis extends to multi-head attention. Linearization at $y_C = 0$ yields a Jacobian that is a sum of products of per-head attention matrices, and approximating $\mathbf{K}^{-1}$ in any such form faces the same Chebyshev barrier. The $\Theta(\sqrt{\kappa})$ depth scaling is intrinsic to the condition number.*

Theorem 18 (Dimension Scaling). *To represent GP posterior predictions on contexts of size n: (a) CNP and ANP require infinite d; (b) TNP requires $d = O(n)$.*

7. Convolutional Neural Processes

ConvCNPs replace finite-dimensional aggregation with functional convolutional representations. Two structural properties distinguish them.

Proposition 19 (Translation Equivariance). *Every ConvCNP satisfies $F(\{(x_i+\tau, y_i)\}, x_t+\tau) = F(\{(x_i, y_i)\}, x_t)$ for all $\tau \in \mathbb{R}^{d_x}$.*

Proposition 20 (Injectivity). *If w has non-vanishing Fourier transform and h is injective, then the convolutional aggregation $C \mapsto (s_C(\cdot), \rho_C(\cdot))$ is injective up to permutation on contexts with distinct locations.*

Thus ConvCNPs avoid the collision problem of Proposition 4, but at the cost that any non-translation-equivariant predictive map lies outside the ConvCNP function class. Proofs appear in Appendix E.

Proposition 21 (Pure ConvCNP Represents Stationary Kernel Smoothers). *For any continuous stationary kernel K, the Nadaraya–Watson estimator $F(C, x_t) = \sum_i K(x_t - x_i)y_i / \sum_i K(x_t - x_i)$ is exactly representable by a pure ConvCNP. However, the GP posterior mean is not representable by any pure ConvCNP (the same factorization barrier as ANPs applies).*

CNN layers enable context–context coupling. On a regular grid, each CNN layer with residual connection implements a Toeplitz matrix iteration $\mathbf{r}^{(\ell)} = (\mathbf{I} + \mathbf{A}_\ell)\mathbf{r}^{(\ell-1)}$, analogous to the TNP update but with Toeplitz structure.

Theorem 22 (ConvCNP Depth for GP Posteriors). *On a regular grid with spacing δ, a ConvCNP with L CNN layers achieves GP posterior approximation error $O(B_k B_y \lambda_{\min}^{-1}((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^L) + O(\delta)$, matching the Chebyshev convergence rate of TNPs.*

A distinctive feature of ConvCNPs is the depth–support tradeoff: on periodic grids, all matrices are circulant and simultaneously diagonalized by the DFT. With unrestricted filter support $p = n$, a single CNN layer suffices (Proposition 55 in Appendix E). With restricted support, the required product $L \cdot \lfloor p/2 \rfloor$ is governed by the trigonometric approximation number of $1/\hat{K}(\omega)$ (Theorem 56 in Appendix E), recovering the $\sqrt{\kappa}$ rate.

Theorem 23 (ConvCNPs and ANPs Are Incomparable). $\mathcal{F}_{\text{ANP}} \not\subseteq \mathcal{F}_{\text{ConvCNP}}$ *(non-stationary kernel smoothers violate translation equivariance) and* $\mathcal{F}_{\text{ConvCNP}} \not\subseteq \mathcal{F}_{\text{ANP}}$ *(ConvCNPs with CNN depth approximate GP posteriors, which are not ANP-representable).*

8. The Expressiveness Hierarchy

Theorem 24 (Expressiveness Hierarchy). *For all* $d \geq 1$ *and* $L \geq 0$*:*

(a) ***Attention branch:*** $\mathcal{F}_{\text{CNP}}^{(d)} \subsetneq \mathcal{F}_{\text{ANP}}^{(d)} \subsetneq \mathcal{F}_{\text{TNP}}^{(1,d)} \subsetneq \mathcal{F}_{\text{TNP}}^{(2,d)} \subsetneq \cdots$

(b) ***Convolutional branch:*** $\mathcal{F}_{\text{CNP}}^{(d)} \subsetneq \mathcal{F}_{\text{ConvCNP}}^{(0)} \subsetneq \mathcal{F}_{\text{ConvCNP}}^{(1)} \subsetneq \mathcal{F}_{\text{ConvCNP}}^{(2)} \subsetneq \cdots$

(c) ***Incomparability:*** $\mathcal{F}_{\text{ANP}}^{(d)} \not\subseteq \mathcal{F}_{\text{ConvCNP}}^{(L)}$ *and* $\mathcal{F}_{\text{ConvCNP}}^{(L)} \not\subseteq \mathcal{F}_{\text{ANP}}^{(d)}$ *for all* L, d*.*

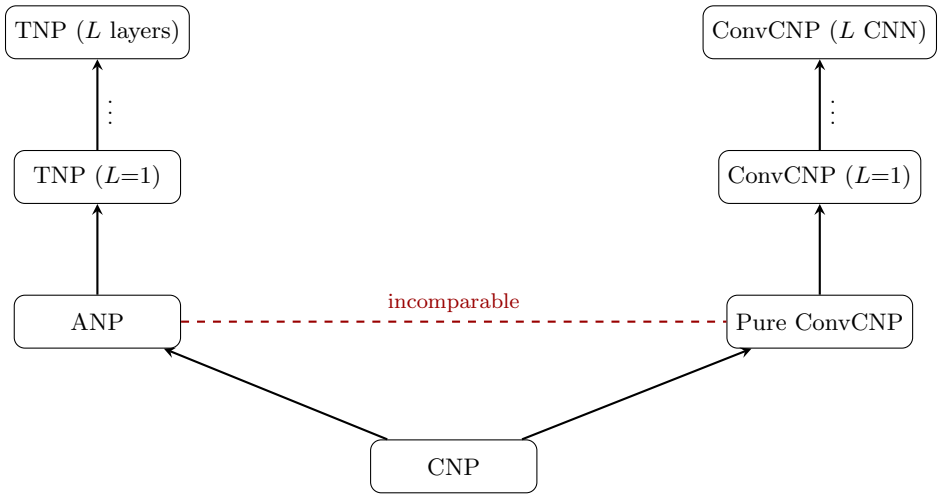

Figure 1. The expressiveness hierarchy of Neural Process architectures. Solid arrows denote strict inclusion ($\subsetneq$). The dashed line marks incomparability. For stationary kernels on regular grids, the ConvCNP branch achieves the same Chebyshev convergence rate as the TNP branch.

The proof assembles the results from previous sections; the full argument appears in Appendix F.

9. Latent Neural Processes

Latent NPs augment the deterministic pathway with a global latent $z \sim q(z|C)$, $z \in \mathbb{R}^k$, yielding $p(y_T|X_T, C) = \int p(y_T|X_T, z)q(z|C)dz$. The encoder $q(z|C)$ inherits the limitations of the underlying architecture: if $C \sim_h C'$ for a latent CNP, then $q(z|C) = q(z|C')$ regardless of decoder expressiveness. Thus the impossibility results for CNPs and ANPs lift directly to their latent variants.

Assuming an arbitrarily powerful encoder, the constraints of finite latent dimension remain:

Theorem 25 (Latent NP Cannot Represent GP Posterior). *For a Gaussian latent NP with latent dimension* k *and linear decoder: (a) Mean matching for all* $y_C \in \mathbb{R}^n$ *requires* $k \geq n$*. (b) Covariance matching at* m *target points requires* $k \geq m$*. (c) Matching for arbitrary target configurations requires* $k = \infty$*.*

Finite-dimensional latent NPs exactly represent the class of rank-k GPs: processes of the form $f(x) = a(x)^\top z + b(x)$ with $z \sim \mathcal{N}(m, S)$. For full-rank kernels, truncating the Mercer expansion gives approximation error governed by the spectral tail $\sum_{j>k} \lambda_j$, with rates depending on kernel smoothness (exponential for RBF, polynomial for Matérn). Full proofs appear in Appendix G.

10. Discussion

Table 1 summarizes our results. The analysis reveals that CNPs are limited to functions of finitely many expected features; ANPs extend this to query-dependent reweightings but still factor across context points; only TNPs can capture global structure via self-attention, with tight depth bounds of $\Theta(\sqrt{\kappa}\log(1/\varepsilon))$. ConvCNPs and ANPs are incomparable, separated by stationarity versus translation equivariance.

Architecture	**Self-Attn/CNN**	**Dim.** d	**Latent** k	**Representable Functions**
CNP	—	any	—	Mean statistics of context
ANP	—	any	—	Query-reweighted statistics
Pure ConvCNP	—	functional	—	Stationary kernel smoothers (exact)
ConvCNP (L CNN)	L CNN layers	functional	—	Stationary degree-L kernel poly.
TNP (L layers)	L self-attn	$O(n)$	—	Degree-L kernel polynomials
TNP (deep)	$\Theta(\sqrt{\kappa}\log\frac{1}{\varepsilon})$	$O(n)$	—	ε-approx GP posterior mean
Latent CNP	—	any	any	Coherent samples from mean stats
Latent ANP	—	any	any	Coherent samples, query-reweighted
Latent TNP	$\Theta(\sqrt{\kappa}\log\frac{1}{\varepsilon})$	$O(n)$	$\geq n$	ε-approx GP posterior dist.

Table 1. Architecture capabilities. Latent variants add coherent sampling but do not circumvent encoder limitations.

Our results provide an answer to a theoretical question open since Garnelo et al. (2018a) introduced NPs regarding what functions these architectures can represent. The analysis reveals a hierarchy. CNPs with mean aggregation are limited to functions of finitely many expected features of the context distribution which is sufficient for low-dimensional parametric families but unable to capture context-context interactions. ANPs extend this to query-dependent reweightings, enabling kernel smoothers and local adaptation but still factoring across context points. Only TNPs with self-attention can capture the global structure needed for GP posteriors, with depth requirements scaling logarithmically in the kernel matrix condition number.

The CNP impossibility follows from a dimension-counting argument. the set of h-equivalent contexts forms an $(n-1)d$-dimensional subspace, so collisions are generic rather than exceptional. The ANP impossibility is more subtle and stems from a factorization constraint because attention weights $\alpha_i \propto f(x_t, x_i, y_i)$ decompose across context points, while GP posterior weights couple all points through $\mathbf{K}^{-1}$. This coupling-versus-factorization distinction, visible even with just two context points, is the fundamental reason self-attention is necessary. The TNP construction exploits the Neumann series $\mathbf{K}^{-1} = \sum_{m=0}^{\infty}(\mathbf{I}-\mathbf{K})^m$, which requires learning an appropriate normalization of the kernel matrix, which is a hidden capacity requirement that may explain why TNPs benefit from careful initialization in practice. Orthogonally to representational capacity, the validity of NP predictions as stochastic processes is governed by conditioning consistency, where the gap for CNPs vanishes as $O(1/n^2)$ in context size (Young, 2026a), while the quantitative cost of amortization decomposes into label contamination, information bottleneck, and encoder sharing terms (Young, 2026b), with the bottleneck decay rates matching the spectral tail bounds of Corollary 66.

Several limitations warrant discussion. Our analysis characterizes representational capacity, not learnability. A function being TNP-representable does not guarantee that gradient descent will find the right parameters, and the sample complexity of learning within each function class remains open. We also assume idealized attention with exact softmax and infinite precision; practical implementations with learned temperatures, finite precision, and approximate attention kernels may exhibit different effective capacity. For standard kernels (e.g. RBF, Matérn), the condition number $\kappa(\mathbf{K})$ typically grows with context size n, so the depth bound $\Theta(\sqrt{\kappa}\log(1/\varepsilon))$ implies that fixed-depth TNPs face representational limitations on large contexts. This is an architectural insight, not a limitation of the analysis as it quantifies the cost of approximating GP posteriors as context sets grow.

Convolutional CNPs (Gordon et al., 2020) replace finite-dimensional aggregation with functional representations, sidestepping the dimension-counting arguments of Section 4. The picture is more nuanced than a simple placement in the hierarchy. The pure ConvCNP (without CNN) sits at the ANP level as its weights on context points factorize, enabling stationary kernel smoothers but not GP posteriors. However, the full ConvCNP with CNN layers accesses TNP-level expressiveness for stationary kernels on regular grids, achieving the same Chebyshev convergence rate $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ per layer (Theorem 22). The structural difference is that CNN iterations are Toeplitz (constant along diagonals), while TNP attention matrices are unconstrained. This makes ConvCNPs and ANPs incomparable rather than nested as each contains functions outside the other (Theorem 23). The cost of irregularity remains an open question.

We also offer some thoughts on practicalities. For few-shot regression on parametric families, CNPs should suffice and attention adds complexity without benefit. For image completion and spatial prediction requiring local adaptation, ANPs provide the right inductive bias. For tasks demanding global coherence such as calibrated uncertainty in Bayesian optimization, consistent identity in face completion, exact GP emulation, TNPs are necessary. For spatially structured tasks with stationary kernels and approximately regular observation grids, which is common in environmental monitoring and climate modeling, ConvCNPs offer an attractive middle ground with TNP-level posterior approximation with the parameter efficiency and equivariance guarantees of convolutional architectures.

References

Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, March 2000. ISSN 1076-9757.

Toby Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall Series in Information and System Sciences. Prentice Hall, 1971. ISBN 9780137531035.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018a. URL https://proceedings.mlr.press/v80/garnelo18a.html.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes, 2018b. URL https://arxiv.org/abs/1807.01622.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, Philadelphia, PA, 2013. doi: 10.1137/1.9781421407944. URL https://epubs.siam.org/doi/abs/10.1137/1.9781421407944.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Skey4eBYPS.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8.

Sharu Theresa Jose and Osvaldo Simeone. Information-theoretic generalization bounds for meta-learning and applications. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010126. URL https://www.mdpi.com/1099-4300/23/1/126.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkE6PjC9KX.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta-learning via sequence modeling. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16569–16594. PMLR, 2022. URL https://proceedings.mlr.press/v162/nguyen22b.html.

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 991–999, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/pentina14.html.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021. URL http://jmlr.org/papers/v22/20-302.html.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine*

Learning Research, pages 9116–9126. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rothfuss21a.html.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1257–1264. MIT Press, 2005. URL https://proceedings.neurips.cc/paper_files/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial & Applied Mathematics, Philadelphia, PA, USA, 2019. ISBN 9781611975932.

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A. Osborne. On the limitations of representing functions on sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6487–6494. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wagstaff19a.html.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1775–1784, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wilson15.html.

Robin Young. On the conditioning consistency gap in conditional neural processes. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856. URL https://openreview.net/forum?id=rLJ5Hm5vbG.

Robin Young. Three costs of amortizing gaussian process inference with neural processes, 2026b. URL https://arxiv.org/abs/2605.21798.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

A. CNP Proofs

Proof of Theorem 3 (CNP Characterization). ($\Rightarrow$) If $F \in \mathcal{F}^{(d)}_{\text{moment}}$, then $F(C, x_t) = \phi(\frac{1}{n}\sum_i \psi(x_i, y_i), x_t)$. Setting $h = \psi$ and $g = \phi$ gives exact representation.

($\Leftarrow$) If F is d-representable, then $F(C, x_t) = g(\bar{h}_C, x_t)$ for some h, g. This is exactly the form of $\mathcal{F}^{(d)}_{\text{moment}}$ with $\psi = h$ and $\phi = g$. □

Proof of Proposition 4 (Existence of Collisions). Consider the map $\Phi : (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^d$ defined by $\Phi(C) = \frac{1}{n}\sum_{i=1}^n h(x_i, y_i)$. The domain has dimension $n(d_x + d_y)$; the codomain has dimension d. When $n(d_x + d_y) > d$, the map cannot be injective, so distinct context sets can have identical representations.

More explicitly, for fixed h-images $(h_1, \ldots, h_n) \in (\mathbb{R}^d)^n$, perturbations $(\delta_1, \ldots, \delta_n)$ satisfying $\sum_i \delta_i = 0$ form a subspace of dimension $(n-1)d$. Since h is continuous and maps from a higher-dimensional space when $n(d_x + d_y) > d$, generic perturbations in $(\mathcal{X} \times \mathcal{Y})^n$ induce such mean-preserving perturbations in the representation space. □

Proof of Theorem 6 (GP Posterior is Not Finitely Representable). The GP posterior mean depends on the inverse Gram matrix $\mathbf{K}^{-1}$, which couples all context points. The weight on y_i in the posterior mean depends on all pairwise kernel values $k(x_i, x_j)$ for $j \neq i$.

By Proposition 4, for any encoder h, there exist distinct context sets C, C' with $\bar{h}_C = \bar{h}_{C'}$. We construct C, C' with identical mean encodings but different x-configurations, hence different Gram matrices $\mathbf{K}(X_C, X_C) \neq \mathbf{K}(X_{C'}, X_{C'})$.

Specifically, with $n = d + 1$ context points, let C have points $\{x_1, \ldots, x_{d+1}\}$ in general position and C' have points $\{x'_1, \ldots, x'_{d+1}\}$ arranged so that $\bar{h}_C = \bar{h}_{C'}$ (possible by the null space argument) but the pairwise distances differ.

For the same target x_t and y-values, the GP posterior means differ:

$$\mu(x_t|C) = k(x_t, X_C)\mathbf{K}(X_C, X_C)^{-1} y_C \neq k(x_t, X_{C'})\mathbf{K}(X_{C'}, X_{C'})^{-1} y_C = \mu(x_t|C').$$

Thus no CNP with representation dimension d can exactly represent the GP posterior on contexts of size $n > d$. □

Proof of Theorem 7 (CNP Approximation Lower Bound). We proceed in four steps.

Step 1: Reduction to linear encoders. Since the target $\mu(x_t|C) = k(x_t, X_C)\mathbf{K}^{-1} y_C$ is linear in y_C and y_C is Gaussian, the optimal encoder is linear. This follows from a standard result in Gaussian rate-distortion theory: for jointly Gaussian $(y_C, T(y_C))$ with T linear, the minimum MSE estimator of $T(y_C)$ given any function $\phi(y_C)$ compressed to d dimensions is achieved by a linear ϕ (see Berger (1971) for example).

For fixed context locations, a CNP encoder $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ reduces to $\bar{h}_C = \frac{1}{n}\sum_{i=1}^n \phi_i(y_i)$ for functions $\phi_i : \mathbb{R} \to \mathbb{R}^d$. Linearity gives $\phi_i(y_i) = a_i y_i$ for some $a_i \in \mathbb{R}^d$, so $\bar{h}_C = A y_C$ for a matrix $A \in \mathbb{R}^{d \times n}$.

Step 2: Whitening. Let $z = \mathbf{K}^{-1/2} y_C \sim \mathcal{N}(0, \mathbf{I}_n)$. Define:

$$\tilde{w}_{x_t} = \mathbf{K}^{-1/2} k(x_t, X_C) \in \mathbb{R}^n$$
$$\tilde{A} = A\mathbf{K}^{1/2} \in \mathbb{R}^{d \times n}$$

Then $\mu(x_t|C) = \tilde{w}_{x_t}^\top z$ and $\bar{h}_C = \tilde{A} z$.

Step 3: Optimal reconstruction. For Gaussian z, the optimal predictor of $\tilde{w}_{x_t}^\top z$ given $\tilde{A} z$ is:

$$\mathbb{E}[\tilde{w}_{x_t}^\top z | \tilde{A} z] = \tilde{w}_{x_t}^\top P_{\tilde{A}} z$$

where $P_{\tilde{A}} = \tilde{A}^\top(\tilde{A}\tilde{A}^\top)^{-1}\tilde{A}$ is the orthogonal projection onto the d-dimensional row space of $\tilde{A}$. The MSE at target x_t is:

$$\mathbb{E}[|\tilde{w}_{x_t}^\top z - \tilde{w}_{x_t}^\top P_{\tilde{A}} z|^2] = \|(\mathbf{I} - P_{\tilde{A}})\tilde{w}_{x_t}\|^2.$$

Step 4: Integrated error. Define the whitened weight matrix $\tilde{W} \in \mathbb{R}^{n\times m}$ with columns $\tilde{w}_{x_1^*}, \ldots, \tilde{w}_{x_m^*}$ for target points $x_1^*, \ldots, x_m^* \sim \nu$. The total MSE is:

$$\sum_{j=1}^{m} \|(\mathbf{I} - P_{\tilde{A}})\tilde{w}_{x_j^*}\|^2 = \|(\mathbf{I} - P_{\tilde{A}})\tilde{W}\|_F^2.$$

This is minimized when $P_{\tilde{A}}$ projects onto the top d left singular vectors of $\tilde{W}$. If $\tilde{W}$ has singular values $\sigma_1 \geq \cdots \geq \sigma_n$ (with $\sigma_i = 0$ for $i > \operatorname{rank}(\tilde{W})$), the minimum error is $\sum_{i=d+1}^{n} \sigma_i^2$ and the total variance is $\sum_{i=1}^{n} \sigma_i^2$.

Under the isotropic second moment assumption:

$$\frac{1}{m}\tilde{W}\tilde{W}^\top = \frac{1}{m}\sum_{j=1}^{m} \tilde{w}_{x_j^*}\tilde{w}_{x_j^*}^\top \to \mathbb{E}_{x_t\sim\nu}[\tilde{w}_{x_t}\tilde{w}_{x_t}^\top] = \alpha\mathbf{I}_n$$

for some $\alpha > 0$. Thus the singular values of $\tilde{W}$ satisfy $\sigma_i^2 \approx \alpha m/n$ for all $i \leq n$, giving:

$$\frac{\sum_{i=d+1}^{n} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2} = \frac{n-d}{n} = 1 - \frac{d}{n}.$$ □

We also record the linear regression example, which illustrates the representation requirements concretely.

Proposition 26 (Linear Regression Requires $d = O(k^2)$). *Let $\mathcal{F}_{\text{linear}}^{(k)}$ be the class of linear predictors with k-dimensional features $\psi : \mathcal{X} \to \mathbb{R}^k$:*

$$F(C, x_t) = \langle \beta(C), \psi(x_t)\rangle, \quad \beta(C) = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \langle \beta, \psi(x_i)\rangle)^2.$$

Then:

(a) $d = \frac{k(k+3)}{2}$ suffices: $\mathcal{F}_{\text{linear}}^{(k)} \subseteq \mathcal{F}_{\text{moment}}^{(k(k+3)/2)}$.

(b) $d = \Omega(k^2)$ is necessary for exact representation.

Proof. (a) The OLS solution is $\beta(C) = (\sum_i \psi(x_i)\psi(x_i)^\top)^{-1}(\sum_i y_i\psi(x_i))$. Define the encoder:

$$h(x, y) = (\operatorname{vech}(\psi(x)\psi(x)^\top), y \cdot \psi(x)) \in \mathbb{R}^{k(k+1)/2+k},$$

Then $\bar{h}_C = (\frac{1}{n}\operatorname{vec}(\sum_i \psi(x_i)\psi(x_i)^\top), \frac{1}{n}\sum_i y_i\psi(x_i))$. The decoder can recover $\beta(C)$ by inverting the (rescaled) Gram matrix and multiplying.

(b) The Gram matrix $\sum_i \psi(x_i)\psi(x_i)^\top$ has $k(k+1)/2$ degrees of freedom (symmetric). Different Gram matrices generically yield different predictors. Thus $d \geq k(k+1)/2 = \Omega(k^2)$. □

B. ANP Proofs

Proof of Theorem 9 (ANPs Represent Kernel Smoothers). Set:

- Value: $v(x, y) = (y, 1) \in \mathbb{R}^{d_y+1}$

- Key: $k(x, y) = \psi(x)$ (independent of y)
- Query: $q(x_t) = \phi(x_t)$

Choose ϕ, ψ such that $q(x_t)^\top k(x_i, y_i) = \log K(x_t, x_i)$. If $\log K$ admits a finite inner product decomposition, this is achieved exactly. For general continuous K on a compact domain, universal approximation (Hornik et al., 1989) gives networks ϕ, ψ with $|q(x_t)^\top k(x_i, y_i) - \log K(x_t, x_i)| \leq \delta$ for any $\delta > 0$, yielding $\varepsilon(\delta)$-approximation of the kernel smoother with $\varepsilon(\delta) \to 0$ as $\delta \to 0$.

In the exact case, the attention weights are:

$$\alpha_i(x_t; C) = \frac{\exp(\log K(x_t, x_i))}{\sum_j \exp(\log K(x_t, x_j))} = \frac{K(x_t, x_i)}{\sum_j K(x_t, x_j)}.$$

The representation is:

$$r_C(x_t) = \sum_i \alpha_i(x_t; C)\,(y_i, 1) = \left(\frac{\sum_i K(x_t, x_i) y_i}{\sum_i K(x_t, x_i)}, 1\right).$$

A linear decoder extracts the first component, which is exactly the kernel smoother. □

Proof of Theorem 10 (ANP Characterization). ($\Rightarrow$) If F is ANP-representable, the attention mechanism gives exactly this form with $s(x_t, x, y) = q(x_t)^\top k(x, y)/\tau$.

($\Leftarrow$) Any such F can be implemented by setting $q(x_t)^\top k(x, y) = s(x_t, x, y) \cdot \tau$ and using an appropriate decoder G. □

Proof of Theorem 12 (GP Posterior Requires Context-Context Coupling). The GP posterior mean is:

$$\mu(x_t|C) = k(x_t, X_C)\,\mathbf{K}(X_C, X_C)^{-1}\,y_C = \sum_{i=1}^{n} w_i(x_t; C)\,y_i$$

where $w_i(x_t; C) = [k(x_t, X_C)\mathbf{K}^{-1}]_i$.

The weight w_i depends on $\mathbf{K}^{-1}$, which couples all context points. Specifically, w_i depends on $k(x_j, x_m)$ for $j, m \neq i$.

ANP attention weights factor as:

$$\alpha_i(x_t; C) = \frac{f(x_t, x_i, y_i)}{\sum_j f(x_t, x_j, y_j)}$$

for some function f depending on query-key pairs independently. The weight on point i depends only on (x_t, x_i, y_i) and the normalizing constant, not on the relationships between other context points.

Explicit counterexample: Consider $n = 2$ context points. The GP weight on point 1 is:

$$w_1 = \frac{k(x_t, x_1)k(x_2, x_2) - k(x_t, x_2)k(x_1, x_2)}{k(x_1, x_1)k(x_2, x_2) - k(x_1, x_2)^2}.$$

This depends on $k(x_1, x_2)$, the kernel value between the two context points. No ANP attention weight can capture this, as $\alpha_1 \propto \exp(q(x_t)^\top k(x_1, y_1))$ involves only the query-point-1 relationship. □

C. TNP Assumptions and Upper Bound Proofs

C.1. Structural Assumptions

Assumption 1 (Position-Based Self-Attention). *Each self-attention layer uses attention weights that depend only on input positions:*

$$\beta_{ij}^{(\ell)} = \frac{\exp(q_s(x_i)^\top k_s(x_j)/\tau)}{\sum_{m=1}^{n} \exp(q_s(x_i)^\top k_s(x_m)/\tau)}$$

where $q_s, k_s : \mathcal{X} \to \mathbb{R}^{d_k}$ are position encoders. We denote the resulting attention matrix by $\tilde{\mathbf{K}} \in \mathbb{R}^{n\times n}$, with $[\tilde{\mathbf{K}}]_{ij} = \beta_{ij}$.

Remark 27. *This assumption can be realized by: (i) using separate position-based attention heads, (ii) concatenating fixed positional encodings that dominate learned representations, or (iii) architectural modifications that enforce position-based routing. The assumption decouples the "routing" structure (which context points attend to which) from the "content" being routed (the value vectors), enabling our polynomial analysis.*

We impose position-based attention to enable clean polynomial analysis of the layer-wise updates. This assumption is relaxed in Appendix D, where we show that the depth lower bound holds even for fully adaptive representation-based attention.

Assumption 2 (Residual Connections). *Self-attention layers use residual connections with learnable value projections:*

$$h_i^{(\ell)} = h_i^{(\ell-1)} + \sum_{j=1}^{n} \beta_{ij}^{(\ell)} W_v^{(\ell)} h_j^{(\ell-1)}$$

where $W_v^{(\ell)} \in \mathbb{R}^{d\times d}$ is the value projection matrix at layer ℓ.

Assumption 3 (Kernel Matrix Conditioning). *The kernel k is positive definite, and for context sets of size n, the Gram matrix $\mathbf{K}$ has eigenvalues $0 < \lambda_{\min} \le \lambda_1 \le \cdots \le \lambda_n \le \lambda_{\max}$. The condition number is $\kappa = \lambda_{\max}/\lambda_{\min}$.*

Assumption 4 (Attention Approximates Normalized Kernel). *The attention matrix $\tilde{\mathbf{K}}$ satisfies $\tilde{\mathbf{K}} = D^{-1}\mathbf{K}$ where $D = \mathrm{diag}(\mathbf{K1})$ is the row-sum normalization. Realistically, the TNP must learn position encoders q_s, k_s achieving this approximation, which is a nontrivial learning problem.*

C.2. Matrix Form and Basic Lemmas

Lemma 28 (Matrix Form of Updates). *Under Assumptions 1 and 2, the layer-ℓ update is:*

$$H^{(\ell)} = H^{(\ell-1)} + \tilde{\mathbf{K}} H^{(\ell-1)} W_v^{(\ell)\top}$$

or in vectorized form:

$$\mathrm{vec}(H^{(\ell)}) = \left(\mathbf{I}_{nd} + W_v^{(\ell)} \otimes \tilde{\mathbf{K}}\right) \mathrm{vec}(H^{(\ell-1)})$$

where $\otimes$ denotes the Kronecker product. For scalar value weights $W_v^{(\ell)} = \alpha_\ell \mathbf{I}_d$:

$$H^{(\ell)} = (\mathbf{I}_n + \alpha_\ell \tilde{\mathbf{K}}) H^{(\ell-1)}.$$

Lemma 29 (One Layer Captures Kernel-Weighted Sums). *Under Assumption 1, if the position encoders q_s, k_s satisfy $q_s(x)^\top k_s(x') = \tau \log K(x, x') + c$ for some kernel K and constant c, then:*

$$\beta_{ij} = \frac{K(x_i, x_j)}{\sum_{m=1}^{n} K(x_i, x_m)}.$$

After one self-attention layer with value function $v : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$:

$$h_i^{(1)} = h_i^{(0)} + \sum_{j=1}^{n} \frac{K(x_i, x_j)}{\sum_m K(x_i, x_m)} v(x_j, y_j).$$

Proof. Direct computation:

$$\beta_{ij} = \frac{\exp(q_s(x_i)^\top k_s(x_j)/\tau)}{\sum_m \exp(q_s(x_i)^\top k_s(x_m)/\tau)} = \frac{\exp(\log K(x_i, x_j) + c)}{\sum_m \exp(\log K(x_i, x_m) + c)} = \frac{K(x_i, x_j)}{\sum_m K(x_i, x_m)}.$$

The representation update follows from the definition. □

Proposition 30 (Gram Matrix Encoding). *With representation dimension $d \geq n$ and initial encoding $h^{(0)}(x_j, y_j) = (e_j, y_j) \in \mathbb{R}^{n+d_y}$ where e_j is the j-th standard basis vector, one self-attention layer can produce representations containing the i-th row of $\tilde{\mathbf{K}}$:*

$$h_i^{(1)} = (e_i + \tilde{\mathbf{K}}_{i\cdot}, y_i + \textit{(weighted sum of } y_j\textit{)})$$

where $\tilde{\mathbf{K}}_{i\cdot} = (\beta_{i1}, \ldots, \beta_{in})$ is the i-th row of the attention matrix.

Proof. Set $W_v^{(1)} = \mathbf{I}$. Then $h_i^{(1)} = h_i^{(0)} + \sum_j \beta_{ij} h_j^{(0)} = (e_i, y_i) + \sum_j \beta_{ij}(e_j, y_j) = (e_i + \tilde{\mathbf{K}}_{i\cdot}, y_i + \sum_j \beta_{ij} y_j)$. □

C.3. Polynomial Computation

Proof of Theorem 13 (Polynomial Representation). By induction. For $L = 1$: $H^{(1)} = (\mathbf{I} + \alpha_1 \tilde{\mathbf{K}}) H^{(0)}$. Assuming the result for $L - 1$:

$$H^{(L)} = (\mathbf{I} + \alpha_L \tilde{\mathbf{K}}) H^{(L-1)} = (\mathbf{I} + \alpha_L \tilde{\mathbf{K}}) \prod_{\ell=1}^{L-1} (\mathbf{I} + \alpha_\ell \tilde{\mathbf{K}}) H^{(0)} = \prod_{\ell=1}^{L} (\mathbf{I} + \alpha_\ell \tilde{\mathbf{K}}) H^{(0)}.$$

Expanding the product, the coefficient of $\tilde{\mathbf{K}}^m$ is the sum over all ways to choose m factors to contribute $\alpha_\ell \tilde{\mathbf{K}}$ (and the remaining $L - m$ factors to contribute $\mathbf{I}$), which is exactly $e_m(\alpha_1, \ldots, \alpha_L)$. □

Corollary 31 (Achievable Coefficient Space). *The set of achievable coefficient vectors $(c_0, c_1, \ldots, c_L) \in \mathbb{R}^{L+1}$ for polynomials $\sum_{m=0}^{L} c_m \tilde{\mathbf{K}}^m$ representable by an L-layer TNP with scalar value weights is:*

$$\mathcal{A}_L = \{(e_0(\alpha), e_1(\alpha), \ldots, e_L(\alpha)) : \alpha \in \mathbb{R}^L\}$$

where $e_0 \equiv 1$. This is an L-dimensional algebraic variety in $\mathbb{R}^{L+1}$, not all of $\mathbb{R}^{L+1}$.

Proof. The map $\alpha \mapsto (e_0(\alpha), \ldots, e_L(\alpha))$ has image determined by Newton's identities relating elementary symmetric polynomials to power sums. Since $e_0 = 1$ always, the image lies in the hyperplane $\{c_0 = 1\}$. Within this hyperplane, the image is the set of coefficient vectors of constant term 1 polynomials with real roots (since $\prod_\ell (1 + \alpha_\ell z) = \sum_m e_m z^m$ has roots $-1/\alpha_\ell$). This is a proper subset of $\mathbb{R}^L$. □

Remark 32 (Non-Scalar Value Matrices). *With general $W_v^{(\ell)} \in \mathbb{R}^{d \times d}$, the achievable polynomial space is larger but the analysis becomes representation-dependent. For a fixed initial representation $H^{(0)}$ and target polynomial $p(\tilde{\mathbf{K}})$, the question becomes whether there exist $W_v^{(1)}, \ldots, W_v^{(L)}$ such that $\prod_\ell (\mathbf{I} + \tilde{\mathbf{K}} W_v^{(\ell)\top}) H^{(0)} = p(\tilde{\mathbf{K}}) H^{(0)}$. This is generically solvable for polynomials of degree $\leq L$ when d is sufficiently large, but a complete characterization requires specifying $H^{(0)}$.*

C.4. Approximating the Kernel Matrix Inverse

Lemma 33 (Neumann Series). *Let $A \in \mathbb{R}^{n\times n}$ with spectral radius $\rho(A) < 1$. Then $(\mathbf{I} - A)^{-1} = \sum_{m=0}^{\infty} A^m$, and the truncated series satisfies:*

$$\left\| (\mathbf{I} - A)^{-1} - \sum_{m=0}^{L-1} A^m \right\| \leq \frac{\rho(A)^L}{1 - \rho(A)}.$$

Proof. Standard result from matrix analysis (Golub and Van Loan, 2013). The truncation error is $\|(\mathbf{I} - A)^{-1}A^L\| = \|(\mathbf{I} - A)^{-1}\|\|A^L\| \leq \frac{1}{1-\rho(A)}\rho(A)^L$. □

Proposition 34 (Inverse via Neumann Series). *Under Assumption 3, setting $A = \mathbf{I} - \mathbf{K}/\lambda_{\max}$:*

(a) The eigenvalues of A lie in $[0, 1 - 1/\kappa]$, so $\rho(A) = 1 - 1/\kappa < 1$.

(b) $\mathbf{K}^{-1} = \frac{1}{\lambda_{\max}}(\mathbf{I} - A)^{-1} = \frac{1}{\lambda_{\max}} \sum_{m=0}^{\infty} A^m$.

(c) The truncated series gives:

$$\left\| \mathbf{K}^{-1} - \frac{1}{\lambda_{\max}} \sum_{m=0}^{L-1} A^m \right\| \leq \frac{\rho^L}{\lambda_{\min}}, \quad \rho = 1 - \frac{1}{\kappa}.$$

Proof. (a) If $\mathbf{K}v = \lambda v$, then $Av = (1 - \lambda/\lambda_{\max})v$. Since $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, the eigenvalues of A lie in $[0, 1 - \lambda_{\min}/\lambda_{\max}] = [0, 1 - 1/\kappa]$.

(b) $\mathbf{K} = \lambda_{\max}(\mathbf{I} - A)$, so $\mathbf{K}^{-1} = \frac{1}{\lambda_{\max}}(\mathbf{I} - A)^{-1}$.

(c) By Lemma 33: $\|(\mathbf{I} - A)^{-1} - \sum_{m=0}^{L-1} A^m\| \leq \frac{\rho^L}{1-\rho} = \frac{\rho^L}{1/\kappa} = \kappa\rho^L$. Dividing by $\lambda_{\max}$: $\|\mathbf{K}^{-1} - \frac{1}{\lambda_{\max}} \sum_{m=0}^{L-1} A^m\| \leq \frac{\kappa\rho^L}{\lambda_{\max}} = \frac{\rho^L}{\lambda_{\min}}$. □

Proposition 35 (Idealized TNP Approximation of GP Posterior). *Suppose a TNP architecture has access to:*

(i) Position-based self-attention with attention matrix equal to the row-normalized kernel $\tilde{\mathbf{K}} = D^{-1}\mathbf{K}$,

(ii) The normalization constants $D_{ii} = \sum_j K(x_i, x_j)$ via the encoding,

(iii) Scalar value weights $\alpha_1, \ldots, \alpha_L$ implementing the truncated Neumann series.

Then the GP posterior mean can be approximated with error $O(\rho^L/\lambda_{\min})$ where $\rho = 1 - 1/\kappa$.

This is an existence result. It establishes that the approximation lies within the representational capacity of TNPs, but does not address whether gradient-based learning can find the required parameters.

Proof. By Proposition 34, the truncated Neumann series $\hat{\mathbf{K}}^{-1} = \frac{1}{\lambda_{\max}} \sum_{m=0}^{L-1} (\mathbf{I} - \mathbf{K}/\lambda_{\max})^m$ satisfies $\|\mathbf{K}^{-1} - \hat{\mathbf{K}}^{-1}\| \leq \rho^L/\lambda_{\min}$.

The approximation error in the posterior mean is:

$$|\mu_{\text{TNP}} - \mu| = |k(x_t, X_C)(\hat{\mathbf{K}}^{-1} - \mathbf{K}^{-1})y_C| \leq \|k(x_t, X_C)\| \cdot \|\hat{\mathbf{K}}^{-1} - \mathbf{K}^{-1}\| \cdot \|y_C\|.$$ □

Corollary 36 (Depth Requirement). *To achieve ε-approximation of the GP posterior mean uniformly over targets x_t with $\|k(x_t, X_C)\| \le B_k$ and observations $\|y_C\| \le B_y$, a TNP requires:*

$$L \ge \frac{\log(B_k B_y/(\varepsilon\lambda_{\min}))}{\log(1/\rho)} = O\left(\kappa \log \frac{B_k B_y}{\varepsilon \lambda_{\min}}\right)$$

self-attention layers, using $\log(1/\rho) \approx 1/\kappa$ for large κ.

Proof of Proposition 14 (Chebyshev Upper Bound). Set

$$\alpha_\ell = -\frac{2}{\lambda_{\max} + \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\cos\theta_\ell}, \quad \theta_\ell = \frac{(2\ell - 1)\pi}{2L}.$$

These are the optimal Chebyshev iteration parameters for inverting a matrix with spectrum in $[\lambda_{\min}, \lambda_{\max}]$. Each α_ℓ is real and negative with $\alpha_\ell \in (-2/\lambda_{\min}, -2/\lambda_{\max})$, so the polynomial $p_L(\lambda) = \prod_\ell (1 + \alpha_\ell \lambda)$ satisfies $p_L(0) = 1$ and has all roots $-1/\alpha_\ell$ in $[\lambda_{\min}, \lambda_{\max}]$.

The residual polynomial $r_L(\lambda) = 1 - \lambda \cdot p_L(\lambda)$ is the degree-L polynomial vanishing at 0 that deviates least from 1 on $[\lambda_{\min}, \lambda_{\max}]$ in the sup-norm. Standard Chebyshev theory (Trefethen, 2019) gives $\|r_L\|_\infty \le 2\rho^L$ where $\rho = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$. The stated bound follows from $\|p_L(\mathbf{K}) - \mathbf{K}^{-1}\| = \|\mathbf{K}^{-1}\| \cdot \|r_L(\mathbf{K})\| \le \lambda_{\min}^{-1} \cdot 2\rho^L$. □

D. TNP Lower Bound Proofs

D.1. Linearization

The GP posterior mean $\mu(x_t|C) = k(x_t, X_C)\mathbf{K}^{-1} y_C$ is linear in y_C. Any TNP approximating this target must itself be approximately linear.

Lemma 37 (Linearization). *Let $F : \mathcal{C} \times \mathcal{X} \to \mathbb{R}$ be a TNP achieving*

$$\sup_{\|y_C\| \le 1} |F(C, x_t) - k(x_t, X_C)\mathbf{K}^{-1} y_C| \le \varepsilon.$$

Assume F is twice continuously differentiable in y_C with $\|\nabla^2_{y_C} F\| \le L_2$ uniformly on $\|y_C\| \le 1$. Then:

$$\left\| \left.\frac{\partial F}{\partial y_C}\right|_{y_C = 0} - k(x_t, X_C)\mathbf{K}^{-1} \right\| \le 2\varepsilon + \frac{L_2}{2}.$$

Proof. Let $M = \left.\frac{\partial F}{\partial y_C}\right|_{y_C = 0}$ and $T = k(x_t, X_C)\mathbf{K}^{-1}$. For any unit vector u, Taylor expansion gives:

$$F(\delta u) = F(0) + \delta \cdot Mu + R(\delta)$$

where $|R(\delta)| \le \frac{L_2}{2}\delta^2$ by the smoothness assumption.

The target satisfies $T(\delta u) := k(x_t, X_C)\mathbf{K}^{-1}(\delta u) = \delta \cdot Tu$ (exactly linear).

By the approximation hypothesis:

$$|F(\delta u) - \delta \cdot Tu| \le \varepsilon \quad \text{for all } |\delta| \le 1.$$

Evaluating at $\delta = 0$: $|F(0)| \le \varepsilon$.

For $\delta \ne 0$:

$$|F(0) + \delta \cdot Mu + R(\delta) - \delta \cdot Tu| \le \varepsilon$$

$$|\delta| \cdot |Mu - Tu| \le \varepsilon + |F(0)| + |R(\delta)| \le 2\varepsilon + \frac{L_2}{2}\delta^2.$$

Setting $\delta = 1$:

$$|Mu - Tu| \leq 2\varepsilon + \frac{L_2}{2}.$$

Since u was an arbitrary unit vector, $\|M - T\| \leq 2\varepsilon + \frac{L_2}{2}$. □

Remark 38 (Smoothness of TNPs). *The smoothness assumption holds for TNPs with standard components. Specifically:*

1. *Softmax attention: $\beta_{ij} = \exp(s_{ij})/\sum_k \exp(s_{ik})$ is C^∞ with derivatives bounded by functions of the scores.*
2. *MLP layers with smooth activations (e.g. GELU, softplus, or sufficiently smooth approximations to ReLU) have bounded second derivatives on compact domains.*
3. *Compositions of C^2 functions with bounded derivatives remain C^2 with bounded derivatives.*

For a TNP with L layers, representation dimension d, and weights bounded by $W_{\max}$, the constant L_2 scales polynomially in these quantities. For the asymptotic depth bounds, we treat L_2 as a fixed constant depending on the architecture but not on ε or κ.

D.2. Jacobian Evolution through Self-Attention

Definition 39 (Jacobian Matrix). *At layer ℓ, define the Jacobian matrix $B^{(\ell)} \in \mathbb{R}^{n\times n}$ (suppressing the representation dimension d) by*

$$B^{(\ell)}_{im} = \left.\frac{\partial h_i^{(\ell)}}{\partial y_m}\right|_{y_C=0}.$$

Lemma 40 (Initial Jacobian). *For an encoder $h^{(0)}(x_i, y_i) = a(x_i) + b(x_i)y_i + O(y_i^2)$, the initial Jacobian is diagonal:*

$$B^{(0)}_{im} = \delta_{im} b(x_i).$$

Theorem 41 (Jacobian Evolution). *Consider a self-attention layer with representation-based attention:*

$$h_i^{(\ell)} = h_i^{(\ell-1)} + \sum_{j=1}^{n} \beta_{ij}^{(\ell)} W_v h_j^{(\ell-1)}$$

where $\beta_{ij}^{(\ell)} = \mathrm{softmax}_j(q(h_i^{(\ell-1)})^\top k(h_j^{(\ell-1)})/\tau)$.

Let $\tilde{\mathbf{K}} = \beta^{(\ell)}|_{y_C=0}$ be the attention matrix evaluated at $y_C = 0$. Then:

$$B^{(\ell)} = B^{(\ell-1)} + \tilde{\mathbf{K}} W_v B^{(\ell-1)} + \Gamma^{(\ell)} B^{(\ell-1)}$$

where $\Gamma^{(\ell)}$ is a matrix depending on X and network parameters, arising from attention gradients.

Proof. Differentiating the layer update with respect to y_m and evaluating at $y_C = 0$:

$$\frac{\partial h_i^{(\ell)}}{\partial y_m} = \frac{\partial h_i^{(\ell-1)}}{\partial y_m} + \sum_j \frac{\partial \beta_{ij}^{(\ell)}}{\partial y_m} W_v h_j^{(\ell-1)} + \sum_j \beta_{ij}^{(\ell)} W_v \frac{\partial h_j^{(\ell-1)}}{\partial y_m}.$$

At $y_C = 0$, the third term becomes $\sum_j \tilde{\mathbf{K}}_{ij} W_v B^{(\ell-1)}_{jm}$.

For the attention gradient term, we use the softmax derivative:

$$\frac{\partial \beta_{ij}}{\partial y_m} = \beta_{ij}\left(\frac{\partial s_{ij}}{\partial y_m} - \sum_k \beta_{ik}\frac{\partial s_{ik}}{\partial y_m}\right)$$

where $s_{ij} = q(h_i)^\top k(h_j)/\tau$.

The score gradient is:

$$\frac{\partial s_{ij}}{\partial y_m} = \frac{1}{\tau}\left[(\nabla_h q|_{h_i})^\top \frac{\partial h_i}{\partial y_m}\cdot k(h_j) + q(h_i)\cdot(\nabla_h k|_{h_j})^\top \frac{\partial h_j}{\partial y_m}\right].$$

At $y_C = 0$, $\frac{\partial h_i^{(0)}}{\partial y_m} = \delta_{im} b(x_i)$, so $\frac{\partial s_{ij}}{\partial y_m}\big|_{y_C=0}$ is nonzero only when $m \in \{i, j\}$.

Collecting terms, the attention gradient contribution has the form $\Gamma^{(\ell)} B^{(\ell-1)}$ where $\Gamma^{(\ell)}$ depends on $\tilde{\mathbf{K}}$, query/key gradients, and initial representations—all functions of X alone. □

Corollary 42 (Polynomial Structure). *After L self-attention layers, the Jacobian $B^{(L)} = \frac{\partial H^{(L)}}{\partial y_C}\big|_{y_C=0}$ has the form:*

$$B^{(L)} = \sum_{\substack{m_1,\ldots,m_k \geq 0 \\ m_1+\cdots+m_k \leq L}} C_0 \tilde{\mathbf{K}}^{m_1} C_1 \tilde{\mathbf{K}}^{m_2} \cdots C_{k-1}\tilde{\mathbf{K}}^{m_k} C_k$$

where each $C_i \in \mathbb{R}^{n\times n}$ depends on context locations X and network parameters, but not on observations y_C.

In particular, $B^{(L)}$ is a matrix polynomial in $\tilde{\mathbf{K}}$ of degree at most $2L$. The factor of 2 arises because the attention gradient terms $\Gamma^{(\ell)}$ involve quadratic products $\tilde{\mathbf{K}}_{ij}\tilde{\mathbf{K}}_{ik}$, contributing up to degree 2 per layer when projected onto the eigenvalue-controlled family.

Proof. By induction on L. The base case $L = 0$ gives $B^{(0)} = C_0$ where $[C_0]_{im} = \delta_{im} b(x_i)$ is diagonal.

For the inductive step, Theorem 41 gives:

$$B^{(\ell)} = (\mathbf{I} + \Gamma^{(\ell)})B^{(\ell-1)} + \tilde{\mathbf{K}} W_v B^{(\ell-1)}$$

where $\Gamma^{(\ell)}$ depends only on X. If $B^{(\ell-1)}$ is a degree-$(\ell-1)$ polynomial in $\tilde{\mathbf{K}}$, then $B^{(\ell)}$ is degree at most ℓ. □

D.3. Spectral Analysis

Definition 43 (Spectral Content). *Let $\{v_1, \ldots, v_n\}$ be the eigenvectors of $\tilde{\mathbf{K}}$ with eigenvalues $\mu_1 \leq \cdots \leq \mu_n$. For any vector $u \in \mathbb{R}^n$, its spectral content is $c = (c_1, \ldots, c_n)$ where $u = \sum_{j=1}^n c_j v_j$.*

Lemma 44 (Quadratic Form Structure). *Let $M = \sum_{m=0}^L C_m \tilde{\mathbf{K}}^m$ be a commutative degree-L matrix polynomial (i.e., C_m commutes with $\tilde{\mathbf{K}}$ for all m). Let $\{v_1, \ldots, v_n\}$ be the orthonormal eigenvectors of $\tilde{\mathbf{K}}$ with eigenvalues $\mu_1 \leq \cdots \leq \mu_n$. Then for any eigenvector v_j:*

$$v_j^\top M v_j = \sum_{m=0}^L (v_j^\top C_m v_j)\mu_j^m,$$

a univariate polynomial in μ_j of degree at most L.

For the general (non-commutative) interleaved form of Corollary 42, the analogous reduction to a univariate polynomial in a single eigenvalue requires restricting to kernel families where all but one eigenvalue is constant, as in Lemma 45.

Proof. Write $\tilde{\mathbf{K}} = \sum_k \mu_k v_k v_k^\top$. Then $\tilde{\mathbf{K}}^m = \sum_k \mu_k^m v_k v_k^\top$, so:

$$u^\top \tilde{\mathbf{K}}^m u = \sum_k \mu_k^m (u^\top v_k)^2.$$

For the full polynomial:

$$u^\top M u = \sum_{m=0}^{L} u^\top C_m \tilde{\mathbf{K}}^m u = \sum_{m=0}^{L} \sum_k \mu_k^m \cdot u^\top C_m v_k \cdot v_k^\top u.$$

Expanding $C_m v_k = \sum_j (v_j^\top C_m v_k) v_j$:

$$u^\top M u = \sum_{m=0}^{L} \sum_{j,k} \mu_k^m (v_j^\top C_m v_k)(u^\top v_j)(u^\top v_k).$$

Setting $Q_{jk}(\mu) = \sum_{m=0}^{L} (v_j^\top C_m v_k) \mu_k^m$ gives the result. Each Q_{jk} has degree at most L in μ_k, hence total degree at most L. □

D.4. Eigenvalue-Controlled Kernel Family

Lemma 45 (Eigenvalue-Controlled Kernel Family). *For any $\kappa > 1$ and $n \geq 2$, there exists a family of positive definite matrices $\{K_t\}_{t \in [0, 1-1/\kappa]}$ such that:*

(i) All row sums equal 1: $K_t \mathbf{1} = \mathbf{1}$, hence $D_t = I$ and $\tilde{K}_t = K_t$

(ii) The minimum eigenvalue is $\mu_1(t) = 1/\kappa + t \in [1/\kappa, 1]$

(iii) The minimum eigenvector v_1 is constant across the family, with $v_1 \perp \mathbf{1}$

(iv) All other eigenvalues equal 1

(v) $\alpha = v_1^\top D_t^{-1} v_1 = 1$ for all t

Proof. Let $\{v_1, \ldots, v_{n-1}\}$ be any orthonormal set in $\mathbf{1}^\perp$, and set $v_n = \mathbf{1}/\sqrt{n}$. Define:

$$K_0 = \frac{1}{\kappa} v_1 v_1^\top + \sum_{j=2}^{n} v_j v_j^\top = I + \left(\frac{1}{\kappa} - 1\right) v_1 v_1^\top.$$

Then K_0 is positive definite with eigenvalue $1/\kappa$ for v_1 and eigenvalue 1 for $v_2, \ldots, v_n$.

Row sums: $K_0 \mathbf{1} = \mathbf{1} + (1/\kappa - 1) v_1 (v_1^\top \mathbf{1}) = \mathbf{1}$ since $v_1 \perp \mathbf{1}$.

For $t \in [0, 1 - 1/\kappa]$, set $K_t = K_0 + t \cdot v_1 v_1^\top$. Then:

- $K_t \mathbf{1} = K_0 \mathbf{1} + t \cdot v_1 (v_1^\top \mathbf{1}) = \mathbf{1}$
- $K_t v_1 = (1/\kappa + t) v_1$
- $K_t v_j = v_j$ for $j \geq 2$

The minimum eigenvalue $1/\kappa + t$ remains below 1 for $t \leq 1 - 1/\kappa$, so v_1 remains the minimum eigenvector throughout. □

D.5. Reduction to Univariate Approximation

Lemma 46 (Reduction to Univariate Approximation). *Let $\mathbf{K}_t$ be the family from Lemma 45 with $t \in [\mu_{\min}, \mu_{\max}]$, and let $M(X)$ be the Jacobian of an L-layer TNP evaluated at $y_C = 0$. Then there exists a univariate polynomial q of degree at most $2L$ such that:*

$$v_1^\top M(\mathbf{K}_t) v_1 = q(t) \quad \textit{for all } t \in [\mu_{\min}, \mu_{\max}].$$

Proof. By Corollary 42, M is a sum of terms of the form $C_0 \tilde{\mathbf{K}}^{m_1} C_1 \cdots C_k$ with $\sum m_i \leq L$, where each C_i depends on X but not on y_C.

For the family $\mathbf{K}_t$ from Lemma 45, we have $\tilde{\mathbf{K}}_t = \mathbf{K}_t$ (since $D_t = I$) with eigenvalues $\mu_1(t) = 1/\kappa + t$ and $\mu_j = 1$ for $j \geq 2$. The eigenvector v_1 is fixed across the family.

Consider any term $C_0 \tilde{\mathbf{K}}_t^{m_1} C_1 \tilde{\mathbf{K}}_t^{m_2} \cdots C_k$. Expanding in the eigenbasis $\{v_1, \ldots, v_n\}$:

$$v_1^\top C_0 \tilde{\mathbf{K}}_t^{m_1} C_1 \cdots C_k v_1 = \sum_{j_0, \ldots, j_k} (v_1^\top C_0 v_{j_0})(v_{j_0}^\top \tilde{\mathbf{K}}_t^{m_1} v_{j_1}) \cdots (v_{j_{k-1}}^\top C_k v_1).$$

Since $\tilde{\mathbf{K}}_t^m v_j = \mu_j^m v_j$, each factor $v_{j_{i-1}}^\top \tilde{\mathbf{K}}_t^{m_i} v_{j_i} = \mu_{j_i}^{m_i} \delta_{j_{i-1}, j_i}$. The sum collapses to paths through eigenvector indices.

For indices $j \geq 2$, we have $\mu_j = 1$, contributing constant factors. Only paths passing through $j = 1$ contribute powers of $\mu_1(t)$. Therefore:

$$v_1^\top M(\mathbf{K}_t) v_1 = \sum_{m=0}^{L} c_m \mu_1(t)^m = q(\mu_1(t))$$

where the coefficients c_m depend on the matrices C_i (hence on network parameters) but not on t. The degree is at most $2L$ since each of the L layers contributes at most degree 2 due to the quadratic attention gradient terms. □

D.6. Chebyshev Barrier

Theorem 47 (Chebyshev Lower Bound). *For any degree-L polynomial p and any interval $[a, b]$ with $0 < a < b$:*

$$\max_{\mu \in [a,b]} \left| p(\mu) - \frac{1}{\mu} \right| \geq \frac{2}{a+b} \rho^L$$

where $\rho = \frac{\sqrt{b/a}-1}{\sqrt{b/a}+1} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ and $\kappa = b/a$ is the condition number.

Proof. This is a classical result from approximation theory. The optimal degree-L polynomial approximation to $1/\mu$ on $[a, b]$ is achieved by appropriately shifted and scaled Chebyshev polynomials, with the stated error bound. See Trefethen (2019). □

D.7. Spectral Relationship

Lemma 48 (Spectral Relationship). *Let $\gamma = d_{\max}/d_{\min}$ where $d_{\max} = \max_i D_{ii}$ and $d_{\min} = \min_i D_{ii}$. Then:*

$$\frac{1}{\gamma} \kappa(\mathbf{K}) \leq \kappa(\tilde{\mathbf{K}}) \leq \gamma \cdot \kappa(\mathbf{K}).$$

Proof. The matrix $\tilde{\mathbf{K}} = D^{-1}\mathbf{K}$ is similar to $\hat{\mathbf{K}} = D^{-1/2}\mathbf{K}D^{-1/2}$:

$$D^{1/2}\tilde{\mathbf{K}}D^{-1/2} = D^{-1/2}\mathbf{K}D^{-1/2} = \hat{\mathbf{K}}.$$

Thus $\tilde{\mathbf{K}}$ and $\hat{\mathbf{K}}$ share eigenvalues, and we analyze the latter via Rayleigh quotients.

Upper bound on $\kappa(\tilde{\mathbf{K}})$: For any unit vector w, setting $u = D^{-1/2}w$ gives:

$$\frac{w^\top \hat{\mathbf{K}} w}{w^\top w} = \frac{u^\top \mathbf{K} u}{u^\top D u} \in \left[\frac{\lambda_{\min}(\mathbf{K})}{d_{\max}}, \frac{\lambda_{\max}(\mathbf{K})}{d_{\min}}\right]$$

where the bounds follow from $d_{\min}\|u\|^2 \le u^\top D u \le d_{\max}\|u\|^2$. Therefore:

$$\kappa(\tilde{\mathbf{K}}) \le \frac{\lambda_{\max}(\mathbf{K})/d_{\min}}{\lambda_{\min}(\mathbf{K})/d_{\max}} = \gamma \cdot \kappa(\mathbf{K}).$$

Lower bound on $\kappa(\tilde{\mathbf{K}})$: Let $v_{\max}$ be the unit eigenvector of $\mathbf{K}$ with eigenvalue $\lambda_{\max}(\mathbf{K})$. Evaluating at $w = D^{1/2}v_{\max}$:

$$\lambda_{\max}(\tilde{\mathbf{K}}) \ge \frac{v_{\max}^\top \mathbf{K} v_{\max}}{v_{\max}^\top D v_{\max}} = \frac{\lambda_{\max}(\mathbf{K})}{v_{\max}^\top D v_{\max}} \ge \frac{\lambda_{\max}(\mathbf{K})}{d_{\max}}.$$

Let $v_{\min}$ be the unit eigenvector of $\mathbf{K}$ with eigenvalue $\lambda_{\min}(\mathbf{K})$. Evaluating at $w = D^{1/2}v_{\min}$:

$$\lambda_{\min}(\tilde{\mathbf{K}}) \le \frac{v_{\min}^\top \mathbf{K} v_{\min}}{v_{\min}^\top D v_{\min}} = \frac{\lambda_{\min}(\mathbf{K})}{v_{\min}^\top D v_{\min}} \le \frac{\lambda_{\min}(\mathbf{K})}{d_{\min}}.$$

Combining:

$$\kappa(\tilde{\mathbf{K}}) = \frac{\lambda_{\max}(\tilde{\mathbf{K}})}{\lambda_{\min}(\tilde{\mathbf{K}})} \ge \frac{\lambda_{\max}(\mathbf{K})/d_{\max}}{\lambda_{\min}(\mathbf{K})/d_{\min}} = \frac{1}{\gamma}\kappa(\mathbf{K}).$$ □

Corollary 49. *When the row-sum ratio $\gamma = O(1)$, we have $\kappa(\tilde{\mathbf{K}}) = \Theta(\kappa(\mathbf{K}))$. Consequently, the depth requirement of $\Theta(\sqrt{\kappa(\tilde{\mathbf{K}})}\log(1/\varepsilon))$ layers for TNP approximation of GP posteriors is equivalent to $\Theta(\sqrt{\kappa(\mathbf{K})}\log(1/\varepsilon))$ in terms of the original kernel matrix condition number.*

D.8. Target Quadratic Form

Lemma 50 (Target Quadratic Form). *Let v_1 be the minimum eigenvector of $\tilde{\mathbf{K}}$. The target value satisfies:*

$$v_1^\top \mathbf{K}^{-1} v_1 = \frac{\alpha}{\mu_1}$$

where $\alpha = v_1^\top D^{-1} v_1 > 0$ and $\mu_1 = \lambda_{\min}(\tilde{\mathbf{K}})$.

Proof. Since $\mathbf{K} = D\tilde{\mathbf{K}}$, we have $\mathbf{K}^{-1} = \tilde{\mathbf{K}}^{-1}D^{-1}$. Thus:

$$v_1^\top \mathbf{K}^{-1} v_1 = v_1^\top \tilde{\mathbf{K}}^{-1} D^{-1} v_1 = \frac{1}{\mu_1} v_1^\top D^{-1} v_1 = \frac{\alpha}{\mu_1}$$

where the second equality uses $\tilde{\mathbf{K}}^{-1}v_1 = \frac{1}{\mu_1}v_1$. Since D^{-1} is positive diagonal and $v_1 \neq 0$, we have $\alpha > 0$. □

D.9. Main Lower Bound

Proof of Theorem 15 (TNP Depth Lower Bound). Step 1: Linearization. By Lemma 37, there exists $M(X)$ with $\|M(X)-\mathbf{K}^{-1}\| \le 2\varepsilon+L_2/2$. For ε sufficiently small relative to the fixed architecture constant L_2, this is $O(\varepsilon)$.

Step 2: Polynomial structure. By Corollary 42, $M(X)$ has the form:

$$M(X) = \sum_{\substack{m_1,\dots,m_k\ge 0\\ m_1+\cdots+m_k\le L}} C_0(X)\tilde{\mathbf{K}}^{m_1}C_1(X)\cdots C_k(X)$$

where each $C_i(X)$ depends on context locations and network parameters, but not on y_C.

Step 3: Restriction to controlled family. We use the family $\{\mathbf{K}_t\}$ from Lemma 45 with $\mu_{\min} = 1/\kappa$ and $\mu_{\max} = 1$. This family satisfies:

- Condition number $\kappa(\mathbf{K}_t) = \kappa$ (ratio of largest to smallest eigenvalue is $1/(1/\kappa) = \kappa$)
- Row-sum ratio $\gamma = 1$ (since $D_t = I$)
- Eigenvector coefficient $\alpha = v_1^\top D_t^{-1} v_1 = \|v_1\|^2 = 1$

Step 4: Univariate reduction. By Lemma 46, for this family:

$$v_1^\top M(\mathbf{K}_t) v_1 = q(\mu_1(t))$$

where $\mu_1(t) = \lambda_{\min}(\tilde{\mathbf{K}}_t)$ and q is a polynomial of degree at most $2L$.

Step 5: Target value. By Lemma 50:

$$v_1^\top \mathbf{K}_t^{-1} v_1 = \frac{\alpha(t)}{\mu_1(t)} \ge \frac{\alpha_0}{\mu_1(t)}.$$

Step 6: Chebyshev barrier. The approximation requirement $\|M - \mathbf{K}_t^{-1}\| \le O(\varepsilon)$ implies:

$$\left| q(\mu_1) - \frac{\alpha(t)}{\mu_1} \right| \le O(\varepsilon) \quad \text{for all } \mu_1 \in [\mu_{\min}, \mu_{\max}].$$

Since $\alpha(t) \ge \alpha_0$ and q must approximate $\alpha(t)/\mu_1 \ge \alpha_0/\mu_1$, Theorem 47 gives:

$$\sup_{\mu_1\in[\mu_{\min},\mu_{\max}]} \left| q(\mu_1) - \frac{\alpha_0}{\mu_1} \right| \ge \frac{2\alpha_0}{\mu_{\min}(1+\kappa)} \cdot \rho^L$$

where $\rho = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$.

Step 7: Solve for L. For the bound $O(\varepsilon)$ to hold:

$$\frac{2\alpha_0}{\mu_{\min}(1+\kappa)} \cdot \rho^L \le O(\varepsilon).$$

Taking logarithms:

$$L \ge \frac{1}{\log(1/\rho)} \log\left(\frac{c\alpha_0}{\varepsilon \mu_{\min}(1+\kappa)} \right).$$

Using $\log(1/\rho) = \log \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} = 2\,\mathrm{arctanh}(1/\sqrt{\kappa}) \approx \frac{2}{\sqrt{\kappa}}$ for large κ, and noting that q has degree at most $2L$:

$$2L \ge \frac{\sqrt{\kappa}}{2} \log\left(\frac{c'\alpha_0}{\varepsilon} \right), \quad \text{hence} \quad L \ge \frac{\sqrt{\kappa}}{4} \log\left(\frac{c'\alpha_0}{\varepsilon} \right)$$

for a constant c' absorbing the $\mu_{\min}$ and $(1+\kappa)$ factors into the logarithm. □

D.10. Dimension Scaling

Proof of Theorem 18 (Dimension Scaling), part (c). The GP posterior mean at target x_t is:

$$\mu(x_t|C) = \sum_{i=1}^{n} w_i(x_t; C) y_i, \quad w_i(x_t; C) = [k(x_t, X_C)\mathbf{K}^{-1}]_i.$$

Encoding strategy: Use initial encoding $h_i^{(0)} = (e_i, y_i, x_i) \in \mathbb{R}^{n+d_y+d_x}$ where e_i is a one-hot index vector.

After L self-attention layers: By Proposition 30 and the polynomial computation theorem, representations can encode sufficient information about $\mathbf{K}$ and its powers to approximate $\mathbf{K}^{-1}y_C$ via the Neumann series.

Cross-attention: With query-dependent attention weights $\alpha_i(x_t) \propto k(x_t, x_i)$, the cross-attention output is:

$$r(x_t) = \sum_i \alpha_i(x_t) h_i^{(L)}.$$

If $h_i^{(L)}$ encodes the i-th component of $\mathbf{K}^{-1}y_C$ (call it z_i), then with $\alpha_i(x_t) = k(x_t, x_i)/\sum_j k(x_t, x_j)$:

$$r(x_t) \approx \frac{\sum_i k(x_t, x_i) z_i}{\sum_j k(x_t, x_j)} = \frac{k(x_t, X_C)\mathbf{K}^{-1}y_C}{\sum_j k(x_t, x_j)}.$$

A decoder with access to $\sum_j k(x_t, x_j)$ (computable via an auxiliary attention head) recovers the unnormalized posterior mean. □

E. ConvCNP Proofs

Proof of Proposition 19 (Translation Equivariance). The functional channels satisfy $\rho_{C+\tau}(x+\tau) = \sum_i w(x+\tau-x_i-\tau) = \rho_C(x)$, and identically for $s_{C+\tau}$. Convolution layers commute with translation: for any filter ϕ and translated function $f_\tau(x) = f(x-\tau)$, $(\phi * f_\tau)(x) = (\phi * f)(x-\tau)$. Pointwise nonlinearities preserve this. By induction over CNN layers, $\tilde{r}_{C+\tau}(x+\tau) = \tilde{r}_C(x)$. □

Proof of Proposition 20 (Injectivity of Convolutional Aggregation). Step 1: Recover locations. Taking Fourier transforms, $\hat{\rho}_C(\xi) = \hat{w}(\xi)\sum_{i=1}^n e^{-2\pi i \xi \cdot x_i}$. Since $\hat{w}(\xi) \neq 0$, we recover the characteristic function of the discrete measure $\mu = \sum_i \delta_{x_i}$, which determines $\{x_1, \ldots, x_n\}$ as a multiset.

Step 2: Recover values. Given the locations, $s_C(x_j) = \sum_{i=1}^n w(x_j - x_i)\, h(y_i)$ for $j = 1, \ldots, n$ is the linear system $\mathbf{s} = W\mathbf{h}$ with $W_{ji} = w(x_j - x_i)$. For positive definite w (e.g. Gaussian) and distinct x_i, W is a positive definite Gram matrix, hence invertible. Injectivity of h then recovers y_i. □

Proof of Proposition 21 (Pure ConvCNP Represents Stationary Kernel Smoothers). Set $w = K$ and $h(y) = (y, 1) \in \mathbb{R}^{d_y+1}$. Then $s_C(x_t) = (\sum_i K(x_t - x_i)y_i,\ \sum_i K(x_t - x_i))$ and $\rho_C(x_t) = \sum_i K(x_t - x_i)$. The decoder $g(s, \rho, x_t) = s_{1:d_y}/s_{d_y+1}$ extracts the kernel smoother exactly. □

Proposition 51 (Pure ConvCNP Cannot Represent GP Posteriors). *For generic positive definite stationary kernels, the GP posterior mean is not representable by any pure ConvCNP.*

Proof. A pure ConvCNP computes:

$$F(C, x_t) = g\left(\sum_{i=1}^{n} \frac{w(x_t - x_i)}{\sum_j w(x_t - x_j)}\, h(y_i),\ \rho_C(x_t),\ x_t\right).$$

The weight $\alpha_i(x_t) = w(x_t - x_i)/\sum_j w(x_t - x_j)$ on context point i depends only on the query–point distance $x_t - x_i$ and the set of all such distances $\{x_t - x_j\}_{j=1}^n$. It does not depend on the inter-context distances $\{x_i - x_j\}_{j \neq i}$.

The GP posterior weight $w_i(x_t; C) = [k(x_t, X_C)\mathbf{K}^{-1}]_i$ depends on the full Gram matrix $\mathbf{K}$, which couples all context points. The same two-point counterexample from Theorem 12 applies: the GP weight on y_1 depends on $k(x_1, x_2)$, which no factorized weight can capture. □

E.1. CNN Layers and Toeplitz Iteration

Proposition 52 (CNN Implements Toeplitz Iteration). *Let context locations $\{x_1, \ldots, x_n\}$ lie on a regular grid with spacing δ, and let $w = K$ for a stationary kernel K. A CNN layer with residual connection and filter ϕ_ℓ:*

$$\tilde{r}^{(\ell)}(x) = \tilde{r}^{(\ell-1)}(x) + (\phi_\ell * \tilde{r}^{(\ell-1)})(x)$$

implements, at the context locations, the matrix update:

$$\mathbf{r}^{(\ell)} = (\mathbf{I} + \mathbf{A}_\ell)\, \mathbf{r}^{(\ell-1)}$$

where $[\mathbf{A}_\ell]_{ij} = \delta^{d_x} \phi_\ell(x_i - x_j)$ is a Toeplitz matrix determined by the filter, up to boundary effects of order $O(\delta)$.

Proof. Evaluating the convolution at a context location x_i:

$$(\phi_\ell * \tilde{r}^{(\ell-1)})(x_i) = \int \phi_\ell(x_i - x')\, \tilde{r}^{(\ell-1)}(x')\, dx'.$$

When $\tilde{r}^{(\ell-1)}$ is concentrated near the context locations (which holds for localized w), the integral is dominated by contributions from neighborhoods of each x_j:

$$(\phi_\ell * \tilde{r}^{(\ell-1)})(x_i) \approx \sum_{j=1}^n \phi_\ell(x_i - x_j) \int_{B_\delta(x_j)} \tilde{r}^{(\ell-1)}(x')\, dx' \approx \sum_{j=1}^n \delta^{d_x}\, \phi_\ell(x_i - x_j)\, \tilde{r}^{(\ell-1)}(x_j).$$

Since $x_i - x_j$ depends only on the grid displacement $i - j$, the matrix $[\mathbf{A}_\ell]_{ij} = \delta^{d_x} \phi_\ell(x_i - x_j)$ is Toeplitz. □

Proof of Theorem 22 (ConvCNP Depth for GP Posteriors). On a regular grid, $\mathbf{K}$ is Toeplitz. The Chebyshev iteration $p_L(\mathbf{K}) = \prod_{\ell=1}^L (\mathbf{I} + \alpha_\ell \mathbf{K})$ uses parameters α_ℓ as in Proposition 14. Each factor $\mathbf{I} + \alpha_\ell \mathbf{K}$ is Toeplitz (the sum of identity and a scaled Toeplitz matrix), so each factor is implementable by a single CNN layer with filter $\phi_\ell(u) = \alpha_\ell K(u)/\delta^{d_x}$ via Proposition 52.

The first error term is the Chebyshev approximation rate from Proposition 14. The $O(\delta)$ term accounts for discretization of the convolution integral and boundary effects; both vanish as $\delta \to 0$ for compactly supported contexts.

After L CNN layers, the representation at each context location x_i encodes the i-th component of $p_L(\mathbf{K}) y_C \approx \mathbf{K}^{-1} y_C$. The readout step recovers the posterior mean $\mu(x_t|C) = k(x_t, X_C)\mathbf{K}^{-1} y_C$ via convolutional cross-attention with filter $w = K$. □

E.2. Depth–Support Tradeoff on Regular Grids

Definition 53 (Trigonometric Approximation Number). *For a continuous function $f : [0, 2\pi] \to \mathbb{R}$, define $\mathcal{N}_\varepsilon(f)$ as the minimum degree D such that there exists a trigonometric polynomial $t(\omega) = \sum_{|j| \leq D} c_j e^{ij\omega}$ satisfying $\sup_{\omega \in [0,2\pi]} |t(\omega) - f(\omega)| \leq \varepsilon$.*

Lemma 54 (CNN Jacobian on Periodic Grids). *Consider a ConvCNP on a periodic grid of size n with L CNN layers, each with filter support at most p and residual connections:*

$$\tilde{r}^{(\ell)}(x) = \tilde{r}^{(\ell-1)}(x) + \sigma\big(\phi_\ell * \tilde{r}^{(\ell-1)}(x) + b_\ell\big), \quad \ell = 1, \dots, L. \tag{13}$$

Write the encoder as $h(y) = h(0) + h'(0)\, y + O(y^2)$. The Jacobian of the full ConvCNP output with respect to y_C, evaluated at $y_C = 0$, is a circulant matrix that factors in the Fourier domain as:

$$\hat{J}(k) = \hat{g}(k) \cdot \prod_{\ell=1}^{L} \big(1 + d_\ell\, \hat{\tau}_\ell(k)\big) \cdot h'(0) \cdot \hat{w}(k), \quad k = 0, \dots, n-1,$$

where:

- *$\hat{w}(k)$ is the DFT of the aggregation filter w,*
- *$\hat{\tau}_\ell(k)$ is the DFT of ϕ_ℓ, a trigonometric polynomial of degree at most $\lfloor p/2 \rfloor$ in $\omega_k = 2\pi k/n$,*
- *$d_\ell \in \mathbb{R}$ is the (spatially constant) activation derivative at layer ℓ,*
- *$\hat{g}(k)$ is the Fourier-domain readout transfer function.*

Proof. We trace the Jacobian through the three stages of the ConvCNP: encoding/aggregation, CNN processing, and readout.

Stage 1: Encoding and aggregation. The signal channel is $s_C(x_i) = \sum_j w(x_i - x_j)\, h(y_j)$. Its Jacobian with respect to y_m is:

$$\frac{\partial s_C(x_i)}{\partial y_m} = w(x_i - x_m)\, h'(y_m).$$

At $y_C = 0$, this becomes $w(x_i - x_m) \cdot h'(0)$, the Toeplitz matrix W with entries $W_{im} = w(x_i - x_m)$, scaled by $h'(0)$. On the periodic grid, W is circulant with DFT $\hat{w}(k)$.

The density channel $\rho_C(x_i) = \sum_j w(x_i - x_j)$ is independent of y_C and hence contributes zero to the Jacobian. Note that ρ_C is constant across grid locations by translation invariance of the periodic grid: $\rho_C(x_i) = \rho_0$ for all i.

The constant component $h(0)$ of the encoder contributes $\sum_j w(x_i - x_j)\, h(0)$, which is also independent of y_C and does not affect the Jacobian. Thus only the linear term $h'(0)\, y$ in the encoder expansion contributes.

Stage 2: CNN processing. At $y_C = 0$, the signal channel vanishes (its y_C-dependent part is zero) and the density channel is the spatial constant ρ_0. The CNN input is therefore spatially uniform at $y_C = 0$.

Consider the ℓ-th layer (13). Its Jacobian with respect to its input, evaluated at $y_C = 0$, is:

$$J^{(\ell)} = I + D_\ell\, T_\ell,$$

where T_ℓ is the circulant matrix of filter ϕ_ℓ and $D_\ell = \mathrm{diag}(\sigma'(\phi_\ell * \tilde{r}^{(\ell-1)}(x_i) + b_\ell)|_{y_C=0})$. Since the input $\tilde{r}^{(\ell-1)}|_{y_C=0}$ is spatially uniform (by induction the base case holds as shown above, and each layer preserves spatial uniformity when the input is spatially uniform), the argument of σ' is the same at every location. Thus $D_\ell = d_\ell I$ for a scalar $d_\ell \in \mathbb{R}$.

The full CNN Jacobian is $J_\Phi = \prod_{\ell=1}^{L}(I + d_\ell T_\ell)$. Since each factor is circulant and products of circulant matrices are circulant, J_Φ is circulant with DFT:

$$\hat{J}_\Phi(k) = \prod_{\ell=1}^{L} (1 + d_\ell\, \hat{\tau}_\ell(k)).$$

Each $\hat{\tau}_\ell(k) = \sum_{|m| \le \lfloor p/2 \rfloor} \phi_\ell(m\delta)\, e^{-2\pi i k m/n}$ is a trigonometric polynomial of degree at most $\lfloor p/2 \rfloor$.

Stage 3: Readout. The readout $g(\tilde{r}_C(x_t), \rho_C(x_t), x_t)$ is evaluated at a single location. On the periodic grid with stationary readout, the Jacobian of the readout with respect to the CNN output is a circulant matrix with DFT $\hat{g}(k)$. (For the readout structure in Proposition 21, where g divides by ρ_C, $\hat{g}(k)$ is the constant $1/\rho_0$.)

Composing the three stages by the chain rule gives the stated factorization. □

Proposition 55 (Full-Support Filters Trivialize the Problem). *On a periodic grid of size n, if each CNN filter has support $p = n$, then a single CNN layer ($L = 1$) suffices: there exist filter coefficients ϕ_1 such that the ConvCNP Jacobian satisfies $\hat{J}(k) = 1/\lambda_k$ for all k, provided $d_1 \neq 0$, $h'(0) \neq 0$, and $\hat{w}(k) \neq 0$ for all k.*

Proof. By Lemma 54, the Jacobian at frequency k is $\hat{J}(k) = \hat{g}(k) \cdot (1 + d_1 \hat{\tau}_1(k)) \cdot h'(0) \cdot \hat{w}(k)$. Setting $\hat{J}(k) = 1/\lambda_k$ and solving:

$$\hat{\tau}_1(k) = \frac{1}{d_1}\left(\frac{1}{\lambda_k \cdot \hat{g}(k) \cdot h'(0) \cdot \hat{w}(k)} - 1\right).$$

This determines $\hat{\tau}_1(k)$ independently at each of the n frequencies. With $p = n$, the DFT $\phi_1 \mapsto (\hat{\tau}_1(0), \ldots, \hat{\tau}_1(n-1))$ is a bijection on $\mathbb{C}^n$, so ϕ_1 exists and is unique. The solution is well-defined provided $d_1 \neq 0$ (nonzero activation derivative at $y_C = 0$, which holds for standard activations like GELU or softplus evaluated at the uniform input), $h'(0) \neq 0$ (nontrivial encoder, which holds for $h(y) = (y, 1)$ since $h'(0) = (1, 0)$), and $\hat{w}(k) \neq 0$ for all k (which holds for positive definite filters such as Gaussians, cf. Proposition 20). □

Theorem 56 (Depth–Support Tradeoff). *Let $\mathbf{K}$ be a circulant positive definite matrix on a periodic grid of size n with eigenvalues $\lambda_k = \hat{K}(\omega_k)$, where $\omega_k = 2\pi k/n$. Let $n > 2L\lfloor p/2 \rfloor$. Any ConvCNP with L CNN layers, each with filter support at most p, achieving ε-approximation of $\mathbf{K}^{-1} y_C$ requires:*

$$L \cdot \lfloor p/2 \rfloor \;\geq\; \mathcal{N}_{O(\varepsilon)}\big(c/\hat{K}\big),$$

where $c = (\hat{g} \cdot h'(0) \cdot \hat{w})^{-1}$ absorbs the encoding and readout, and $\mathcal{N}_\varepsilon$ is as in Definition 53.

Proof. Step 1: Linearization. The target $T(y_C) = \mathbf{K}^{-1} y_C$ is linear. By the same argument as Lemma 37 (applied to the ConvCNP as a map $\mathbb{R}^n \to \mathbb{R}^n$), ε-approximation of T implies that the Jacobian $M = \partial F / \partial y_C|_{y_C = 0}$ satisfies $\|M - \mathbf{K}^{-1}\| \leq O(\varepsilon)$, where the constant absorbs the smoothness bound $L_2/2$ which depends on the architecture but not on ε or κ.

Step 2: Fourier structure. By Lemma 54, M is circulant with DFT:

$$\hat{M}(k) = \hat{g}(k) \cdot \prod_{\ell=1}^{L} (1 + d_\ell \, \hat{\tau}_\ell(k)) \cdot h'(0) \cdot \hat{w}(k).$$

Define $q(\omega) = \prod_{\ell=1}^{L} (1 + d_\ell \, \hat{\tau}_\ell(\omega))$, which is a trigonometric polynomial of degree at most $D = L \cdot \lfloor p/2 \rfloor$. Then $\hat{M}(k) = \hat{g}(\omega_k) \cdot q(\omega_k) \cdot h'(0) \cdot \hat{w}(\omega_k)$.

Step 3: Approximation requirement. Since M and $\mathbf{K}^{-1}$ are both circulant, $\|M - \mathbf{K}^{-1}\| = \max_k |\hat{M}(k) - 1/\lambda_k|$. The approximation bound from Step 1 gives:

$$\left| \hat{g}(\omega_k) \cdot q(\omega_k) \cdot h'(0) \cdot \hat{w}(\omega_k) - \frac{1}{\lambda_k} \right| \leq O(\varepsilon) \quad \text{for all } k = 0, \ldots, n-1.$$

Rearranging, the trigonometric polynomial $r(\omega) = \hat{g}(\omega) \cdot h'(0) \cdot \hat{w}(\omega) \cdot q(\omega)$ of degree at most $D + D_0$ (where D_0 is the degree contribution from $\hat{g}$ and $\hat{w}$, a fixed constant depending on the kernel and readout) satisfies:

$$\left| r(\omega_k) - \frac{1}{\hat{K}(\omega_k)} \right| \leq O(\varepsilon) \quad \text{for all } k.$$

Step 4: From grid to uniform. The trigonometric polynomial $r(\omega) - 1/\hat{K}(\omega)$ has degree at most $D + D_0 + D_K$, where D_K is the degree of the numerator of $r(\omega)\hat{K}(\omega) - 1$ expressed as a ratio of trigonometric polynomials. A trigonometric polynomial of degree N with $|t(\omega_k)| \leq O(\varepsilon)$ at n equispaced points satisfies $\|t\|_\infty \leq O(\varepsilon)$ provided $n > 2N$ (since a trigonometric polynomial of degree N is determined by $2N + 1$ equispaced samples). The condition $n > 2L\lfloor p/2 \rfloor$ ensures this (absorbing the fixed contribution D_0 for n sufficiently large relative to D_0).

Thus r is a trigonometric polynomial of degree at most $D + D_0$ that uniformly $O(\varepsilon)$-approximates $1/\hat{K}(\omega)$ on $[0, 2\pi]$. By Definition 53, $D + D_0 \geq \mathcal{N}_{O(\varepsilon)}(1/\hat{K})$. Since D_0 is a fixed constant, $D = L \cdot \lfloor p/2 \rfloor \geq \mathcal{N}_{O(\varepsilon)}(1/\hat{K}) - D_0$, giving $L \cdot \lfloor p/2 \rfloor \geq \mathcal{N}_{O(\varepsilon)}(c/\hat{K})$ as stated. □

Corollary 57 (Recovering the $\sqrt{\kappa}$ Rate). *Fix filter support $p = O(\ell_K/\delta)$, where ℓ_K is the kernel lengthscale. If the spectral density $\hat{K}(\omega)$ satisfies $\hat{K}_{\min} \leq \hat{K}(\omega) \leq \hat{K}_{\max}$ with $\kappa = \hat{K}_{\max}/\hat{K}_{\min}$, then:*

$$L \geq \Omega\left(\frac{\sqrt{\kappa}\,\log(1/\varepsilon)}{p}\right).$$

In particular, for the filter support used in Theorem 22, the depth requirement matches the Chebyshev upper bound up to the factor p.

Proof. The function $1/\hat{K}(\omega)$ takes values in $[1/\hat{K}_{\max}, 1/\hat{K}_{\min}]$. By the substitution $\mu = \hat{K}(\omega)$, approximating $1/\hat{K}(\omega)$ in the trigonometric sense reduces to approximating $1/\mu$ on the interval $[\hat{K}_{\min}, \hat{K}_{\max}]$.

More precisely, let $t(\omega)$ be any trigonometric polynomial satisfying $|t(\omega) - 1/\hat{K}(\omega)| \leq \varepsilon$ uniformly. Define $p(\mu) = t(\hat{K}^{-1}(\mu))$ on any monotone branch of $\hat{K}$. Then $|p(\mu) - 1/\mu| \leq \varepsilon$ on $[\hat{K}_{\min}, \hat{K}_{\max}]$. The degree of t is at least the degree needed for p, which by the classical Chebyshev barrier (Theorem 47) is $\Omega(\sqrt{\kappa}\,\log(1/\varepsilon))$. (The substitution can increase degree by at most the factor $\lfloor q/2 \rfloor$, the trigonometric degree of $\hat{K}$ itself, which is a fixed constant depending on the kernel support.)

Theorem 56 then gives $L \cdot \lfloor p/2 \rfloor \geq \Omega(\sqrt{\kappa}\,\log(1/\varepsilon))$, yielding the stated bound. □

Remark 58 (Non-Grid Contexts). *For irregular context locations, $\mathbf{K}$ is not circulant and the Fourier diagonalization of Lemma 54 breaks down as the CNN layers still produce Toeplitz (circulant) Jacobians, but the target $\mathbf{K}^{-1}$ is no longer circulant. The depth–support tradeoff ceases to apply in its current form, and the relevant question becomes how well circulant matrices can approximate the non-circulant iterates needed for $\mathbf{K}^{-1}$. This "circulant approximation gap" depends on the geometry of context point configurations in ways not captured by the condition number alone, and its characterization remains open.*

E.3. Incomparability with ANPs

Proof of Theorem 23 (ConvCNPs and ANPs Are Incomparable). $\mathcal{F}_{\text{ANP}} \not\subseteq \mathcal{F}_{\text{ConvCNP}}$**:** Let $\sigma : \mathcal{X} \to \mathbb{R}_+$ be non-constant and define the non-stationary kernel smoother:

$$F(C, x_t) = \frac{\sum_i \sigma(x_t)\sigma(x_i)K(x_t - x_i)\,y_i}{\sum_i \sigma(x_t)\sigma(x_i)K(x_t - x_i)}.$$

This is ANP-representable by Theorem 9 with non-stationary kernel $\tilde{K}(x, x') = \sigma(x)\sigma(x')K(x - x')$. However, F is not translation equivariant: shifting all inputs by τ replaces $\sigma(x_i)$ with $\sigma(x_i + \tau) \neq \sigma(x_i)$ generically. Since every ConvCNP is translation equivariant (Proposition 19), $F \notin \mathcal{F}_{\text{ConvCNP}}$.

$\mathcal{F}_{\text{ConvCNP}} \not\subseteq \mathcal{F}_{\text{ANP}}$**:** For a stationary kernel, the GP posterior mean is translation equivariant ($K(x + \tau, x' + \tau) = K(x - x')$ implies invariance of $\mathbf{K}$ under joint translation). By Theorem 22, a ConvCNP with sufficient CNN depth approximates it to arbitrary precision on grid contexts. By Theorem 12, it is not ANP-representable. □

F. Hierarchy Proof

Proof of Theorem 24 (Expressiveness Hierarchy). **Part (a): Attention branch.**

$\mathcal{F}_{\text{CNP}}^{(d)} \subseteq \mathcal{F}_{\text{ANP}}^{(d)}$*:* Set uniform attention weights $\alpha_i = 1/n$ (achieved by constant query and key functions).

$\mathcal{F}_{\text{CNP}}^{(d)} \neq \mathcal{F}_{\text{ANP}}^{(d)}$*:* Kernel smoothers are in $\mathcal{F}_{\text{ANP}}^{(d)}$ (Theorem 9) but not in $\mathcal{F}_{\text{CNP}}^{(d)}$. To see the latter, note that a CNP computes $F(C, x_t) = g(\bar{h}_C, x_t)$ where $\bar{h}_C$ is independent of x_t. For the Gaussian kernel smoother with $n = 2$ points, configurations $C = \{(0,0),(1,1)\}$ and $C' = \{(0.25,0),(0.75,1)\}$ can satisfy $\bar{h}_C = \bar{h}_{C'}$ for appropriate h, yet the kernel smoother outputs differ: at $x_t = 0$, configuration C weights the points roughly equally while C' strongly favors the nearby point $(0.25, 0)$.

$\mathcal{F}_{\text{ANP}}^{(d)} \subseteq \mathcal{F}_{\text{TNP}}^{(1,d)}$*:* An ANP is a TNP with $L = 0$ self-attention layers. With $L = 1$ and $W_v^{(1)} = 0$, we recover ANP.

$\mathcal{F}_{\text{ANP}}^{(d)} \neq \mathcal{F}_{\text{TNP}}^{(1,d)}$*:* GP posteriors are in $\mathcal{F}_{\text{TNP}}^{(L,d)}$ for sufficient L (Proposition 35) but not in $\mathcal{F}_{\text{ANP}}^{(d)}$ (Theorem 12).

$\mathcal{F}_{\text{TNP}}^{(L,d)} \subsetneq \mathcal{F}_{\text{TNP}}^{(L+1,d)}$*:* The inclusion follows by adding a layer with $W_v^{(L+1)} = 0$. We prove strictness under Assumptions 1–2 with scalar value weights; the global lower bound of Theorem 15 establishes the corresponding result for representation-based attention at the coarser granularity of $\Theta(\sqrt{\kappa}\log(1/\varepsilon))$ total layers.

An L-layer TNP with scalar value weights computes $p_L(\tilde{\mathbf{K}})H^{(0)}$ where $p_L(x) = \prod_{\ell=1}^{L}(1+\alpha_\ell x)$. The achievable set $\mathcal{P}_L = \{p_L : \alpha_1, \ldots, \alpha_L \in \mathbb{R}\}$ satisfies $\mathcal{P}_L \subset \text{Poly}_L$, the space of polynomials of degree at most L.

Let $\mathbf{K}$ have condition number $\kappa > 1$ with eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$. The GP posterior requires $\mathbf{K}^{-1}$, which acts as $1/\lambda$ on each eigenspace. The minimax error for degree-L polynomial approximation to $1/x$ on $[\lambda_{\min}, \lambda_{\max}]$ is

$$E_L^* = \inf_{p \in \text{Poly}_L} \sup_{x \in [\lambda_{\min}, \lambda_{\max}]} |p(x) - 1/x| = \Theta\left(\frac{\rho^L}{\lambda_{\min}}\right)$$

where $\rho = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$; see Trefethen (2019). Since $\mathcal{P}_L \subset \text{Poly}_L$:

$$\inf_{p \in \mathcal{P}_L} \|p(\mathbf{K}) - \mathbf{K}^{-1}\| \geq E_L^*.$$

For the upper bound at depth $L + 1$, Proposition 14 provides explicit real parameters $\alpha_1, \ldots, \alpha_{L+1}$ such that $p_{L+1}(\mathbf{K}) = \prod_{\ell=1}^{L+1}(\mathbf{I} + \alpha_\ell \mathbf{K})$ satisfies

$$\|p_{L+1}(\mathbf{K}) - \mathbf{K}^{-1}\| \leq \frac{2}{\lambda_{\min}}\rho^{L+1}.$$

Combining bounds:

$$\frac{\inf_{L\text{-layer}} \|p(\mathbf{K}) - \mathbf{K}^{-1}\|}{\inf_{(L+1)\text{-layer}} \|p(\mathbf{K}) - \mathbf{K}^{-1}\|} \geq \frac{E_L^*}{C\rho^{L+1}/\lambda_{\min}} = \Theta(\rho^{-1}) > 1.$$

Thus the GP posterior (restricted to accuracy ε with $E_{L+1}^* < \varepsilon < E_L^*$) lies in $\mathcal{F}_{\text{TNP}}^{(L+1,d)} \setminus \mathcal{F}_{\text{TNP}}^{(L,d)}$.

Part (b): Convolutional branch.

$\mathcal{F}_{\text{CNP}}^{(d)} \subsetneq \mathcal{F}_{\text{ConvCNP}}^{(0)}$*:* Every CNP function is a pure ConvCNP function: set $w = \mathbf{1}$ (constant filter) to recover mean aggregation $s_C(x_t) = \sum_i h(y_i)$ with $\rho_C(x_t) = n$, giving $F(C, x_t) = g(\frac{1}{n}\sum_i h(y_i), x_t)$. Strictness: stationary kernel smoothers are in $\mathcal{F}_{\text{ConvCNP}}^{(0)}$ (Proposition 21) but not in $\mathcal{F}_{\text{CNP}}^{(d)}$ for any

finite d, since the kernel smoother weight on y_i depends on x_t via $K(x_t - x_i)$, which the query-independent CNP representation cannot capture.

$\mathcal{F}^{(0)}_{\text{ConvCNP}} \subsetneq \mathcal{F}^{(1)}_{\text{ConvCNP}}$*:* The inclusion follows by setting the CNN filter to zero. Strictness: consider the one-hop kernel-weighted sum

$$F(C, x_t) = \frac{\sum_i K(x_t - x_i)\left[\sum_j K(x_i - x_j)\, y_j\right]}{\sum_i K(x_t - x_i)}$$

which depends on inter-context distances $K(x_i - x_j)$. A single CNN layer computes this (Proposition 52), but a pure ConvCNP cannot represent it: the weight on y_j in the pure ConvCNP factors as $\alpha_j(x_t) = w(x_t - x_j)/\sum_m w(x_t - x_m)$, which is independent of all other context locations $\{x_i\}_{i \neq j}$. The one-hop sum assigns y_j a weight proportional to $\sum_i K(x_t - x_i)K(x_i - x_j)$, which couples x_j to every other context point through the intermediate summation.

$\mathcal{F}^{(L)}_{\text{ConvCNP}} \subsetneq \mathcal{F}^{(L+1)}_{\text{ConvCNP}}$*:* The same Chebyshev argument as in Part (a) applies. On regular grids, L CNN layers compute Toeplitz polynomials of degree L in $\mathbf{K}$ (Proposition 52). The set of achievable Toeplitz polynomials at degree L is contained in Poly_L, so the minimax barrier E^*_L applies. At depth $L+1$, the Chebyshev construction of Theorem 22 achieves error $C\rho^{L+1}/\lambda_{\min}$, giving the same separation ratio $\Theta(\rho^{-1}) > 1$.

Part (c): Incomparability.

$\mathcal{F}^{(d)}_{\text{ANP}} \not\subseteq \mathcal{F}^{(L)}_{\text{ConvCNP}}$*:* Non-stationary kernel smoothers are ANP-representable (Theorem 9) but not ConvCNP-representable at any depth, since every ConvCNP is translation equivariant and non-stationary kernel smoothers generically violate translation equivariance.

$\mathcal{F}^{(L)}_{\text{ConvCNP}} \not\subseteq \mathcal{F}^{(d)}_{\text{ANP}}$*:* For $L \geq 1$, ConvCNPs with CNN layers can approximate GP posteriors for stationary kernels on regular grids (Theorem 22). GP posteriors are not ANP-representable (Theorem 12). For $L = 0$, stationary kernel smoothers are exactly representable by pure ConvCNPs (Proposition 21) but only approximately representable by ANPs (Theorem 9), giving a strict separation in the exact representation sense. □

G. Latent NP Proofs

Proposition 59 (Encoder Bottleneck). *For a latent CNP with encoder $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, if $C \sim_h C'$ (i.e., $\bar{h}_C = \bar{h}_{C'}$), then:*

$$q(z|C) = q(z|C') \quad \textit{and hence} \quad p(y_T|X_T, C) = p(y_T|X_T, C')$$

for all target configurations X_T, regardless of decoder expressiveness.

Proof. The latent distribution depends on C only through r_C. If $r_C = r_{C'}$, then $q(z|C) = q(z|C')$. The predictive distribution $p(y_T|X_T, C) = \int p(y_T|X_T, z)q(z|C)dz$ then coincides for C and C'. □

G.1. Mean Bottleneck

Assumption 5 (Gaussian Latent NP). *The latent distribution is Gaussian: $q(z|C) = \mathcal{N}(m(C), S(C))$ with $m : \mathcal{C} \to \mathbb{R}^k$ and $S : \mathcal{C} \to \mathbb{R}^{k\times k}_{++}$. The decoder is linear: $f(x, z) = a(x)^\top z + b(x)$ with $a : \mathcal{X} \to \mathbb{R}^k$ and $b : \mathcal{X} \to \mathbb{R}$.*

Under this assumption, the predictive distribution at targets $X_T = (x_{t_1}, \ldots, x_{t_m})$ is Gaussian with:

$$\mathbb{E}[y_T|X_T, C] = A(X_T)m(C) + b(X_T) \tag{14}$$

$$\text{Cov}(y_T|X_T, C) = A(X_T)S(C)A(X_T)^\top + \sigma^2 I \tag{15}$$

where $A(X_T) \in \mathbb{R}^{m\times k}$ has rows $a(x_{t_i})^\top$ and $b(X_T) = (b(x_{t_1}), \ldots, b(x_{t_m}))^\top$.

Proof of Theorem 25, part (a) (Mean Bottleneck). Fix context locations X_C with n points in general position for a universal kernel k. Define $\phi(x_t) = \mathbf{K}^{-1}k(X_C, x_t) \in \mathbb{R}^n$, so the GP posterior mean is $\phi(x_t)^\top y_C$.

Step 1: Construct target points with independent weight vectors. For a universal kernel, the vectors $\{\phi(x_t) : x_t \in \mathcal{X}\}$ span $\mathbb{R}^n$. Choose n target points $x_{t_1}, \ldots, x_{t_n}$ such that $\Phi = [\phi(x_{t_1}), \ldots, \phi(x_{t_n})]^\top \in \mathbb{R}^{n\times n}$ is invertible.

Step 2: Analyze the composite map. Define $F : \mathbb{R}^k \to \mathbb{R}^n$ by $F(z) = (f(x_{t_1}, z), \ldots, f(x_{t_n}, z))$. The matching condition requires:

$$F(g(X_C, y_C)) = \Phi y_C \quad \forall y_C \in \mathbb{R}^n.$$

Step 3: Dimension argument. Since Φ is invertible, the map $y_C \mapsto \Phi y_C$ is a bijection on $\mathbb{R}^n$. Thus $F \circ g : \mathbb{R}^n \to \mathbb{R}^n$ is surjective.

The image of $g : \mathbb{R}^n \to \mathbb{R}^k$ has dimension at most $\min(n, k)$. For the composite $F \circ g$ to be surjective onto $\mathbb{R}^n$, we need $\dim(\mathrm{im}(g)) \geq n$, hence $k \geq n$.

The stochastic case reduces to the deterministic case by considering the low-variance limit $q(z|C) = \mathcal{N}(m(C), \sigma^2 I)$ as $\sigma \to 0$. □

G.2. Covariance Bottleneck

Theorem 60 (Covariance Rank Bound). *Under Assumption 5, the predictive covariance satisfies:*

$$\mathrm{rank}\left(\mathrm{Cov}(y_T|X_T, C) - \sigma^2 I\right) \leq k$$

for any target configuration X_T and any context C.

Proof. From (15), $\mathrm{Cov}(y_T|X_T, C) - \sigma^2 I = A(X_T)S(C)A(X_T)^\top$. This matrix has rank at most $\min(m, k, \mathrm{rank}(S(C))) \leq k$. □

Theorem 61 (GP Posterior Covariance Rank). *For a universal kernel k, the GP posterior covariance matrix at m target points $x_{t_1}, \ldots, x_{t_m}$ distinct from the context locations X_C has rank m generically.*

Proof. The GP posterior covariance is:

$$\tilde{\Sigma}_T = K_{TT} - K_{TC}\mathbf{K}^{-1}K_{CT}$$

where $K_{TT} = [k(x_{t_i}, x_{t_j})]_{ij} \in \mathbb{R}^{m\times m}$, $K_{TC} = [k(x_{t_i}, x_j)]_{ij} \in \mathbb{R}^{m\times n}$, and $K_{CT} = K_{TC}^\top$.

This is the Schur complement of $\mathbf{K}$ in the joint covariance matrix:

$$\begin{pmatrix} \mathbf{K} & K_{CT} \\ K_{TC} & K_{TT} \end{pmatrix}.$$

For a universal kernel, this joint matrix is positive definite when all $n + m$ points are distinct. The Schur complement of a positive definite block in a positive definite matrix is positive definite. Thus $\tilde{\Sigma}_T \succ 0$, which implies $\mathrm{rank}(\tilde{\Sigma}_T) = m$. □

Corollary 62 (Covariance Mismatch). *A Gaussian latent NP with latent dimension k and linear decoder cannot match the GP posterior covariance at $m > k$ target points.*

Proof. The GP posterior covariance (minus observation noise) has rank m by Theorem 61. The latent NP predictive covariance (minus observation noise) has rank at most k by Theorem 60. For $m > k$, exact matching is impossible. □

Proof of Theorem 25 (Latent NP Cannot Represent GP Posterior). Part (a) is proven above (Mean Bottleneck). Part (b) is Corollary 62. Part (c) follows from (b) since we can choose arbitrarily many target points. □

G.3. What Latent NPs Can Represent

Definition 63 (Finite-Rank GP). *A GP has rank k if its covariance kernel admits the representation:*

$$\tilde{k}(x, x') = \sum_{i,j=1}^{k} S_{ij} a_i(x) a_j(x') = a(x)^\top S\, a(x')$$

for some $a : \mathcal{X} \to \mathbb{R}^k$ and positive semidefinite $S \in \mathbb{R}^{k\times k}$.

Theorem 64 (Latent NP Function Class). *A Gaussian latent NP with latent dimension k and linear decoder exactly represents the class of stochastic processes:*

$$\mathcal{F}^{(k)}_{\text{latent}} = \left\{ f(x) = a(x)^\top z + b(x) : z \sim \mathcal{N}(m, S),\ a : \mathcal{X} \to \mathbb{R}^k,\ b : \mathcal{X} \to \mathbb{R},\ m \in \mathbb{R}^k,\ S \in \mathbb{R}^{k\times k}_{++} \right\}$$

where m and S may depend on the context C.

This is the class of GPs with rank at most k.

Proof. ($\Rightarrow$) A Gaussian latent NP with linear decoder $f(x, z) = a(x)^\top z + b(x)$ and latent $z|C \sim \mathcal{N}(m(C), S(C))$ induces:

$$\mathbb{E}[f(x)|C] = a(x)^\top m(C) + b(x)$$
$$\text{Cov}(f(x), f(x')|C) = a(x)^\top S(C) a(x')$$

which is a rank-k GP.

($\Leftarrow$) Any rank-k GP with mean $\mu(x) = a(x)^\top m + b(x)$ and covariance $\tilde{k}(x, x') = a(x)^\top S a(x')$ is realized by setting the latent distribution to $\mathcal{N}(m, S)$ and decoder to $f(x, z) = a(x)^\top z + b(x)$. □

Corollary 65 (Representable Function Families). *The class $\mathcal{F}^{(k)}_{\text{latent}}$ includes:*

(a) Bayesian linear regression with k basis functions: $f(x) = \sum_{j=1}^{k} z_j \phi_j(x)$.

(b) GPs with degenerate (finite-rank) kernels.

(c) Any k-parameter function family with linear parameter dependence and Gaussian parameter posterior.

Corollary 66 (Spectral Decay Rates). *For common kernels on $[0, 1]^d$:*

*(a) **RBF kernel:** $\lambda_j \sim e^{-cj^{2/d}}$, giving error $O(e^{-c'k^{2/d}})$.*

*(b) **Matérn-ν kernel:** $\lambda_j \sim j^{-(2\nu+d)/d}$, giving error $O(k^{-(2\nu)/d})$.*

*(c) **Polynomial kernel of degree p:** $\lambda_j = 0$ for $j > \binom{p+d}{d}$, giving zero error for $k \geq \binom{p+d}{d}$.*

Remark 67 (Nonlinear Decoders). *For nonlinear decoders $f(x, z)$, the covariance structure is more complex:*

$$\text{Cov}(f(x_t, z), f(x_{t'}, z)|C) = \mathbb{E}_z[f(x_t, z) f(x_{t'}, z)] - \mathbb{E}_z[f(x_t, z)]\mathbb{E}_z[f(x_{t'}, z)].$$

With sufficiently expressive f, the effective rank can exceed k. However, the mutual information bound $I(y_T; C|X_T) \leq I(z; C) \leq H(z)$ still constrains the total information about C that can flow through a finite-dimensional latent. Formalizing this requires rate-distortion arguments beyond our current scope.