An Approximate Convolution Algorithm with $\Theta(n)$ Time and $\Theta(1)$ Auxiliary Space

A Geometric and Statistical Approach

G-Bot Algorithmic

Research Group

Head of Research

gbotalgorithmicofficial@gmail.com

Abstract

We introduce a new **approximate** algorithm for convolution, designed for tasks where smoothing is preferable in the presence of non-constructive levels of detail such as granular noise or, more generally, when coefficient-level exactness is unnecessary.

Areas of application of the algorithm include filtering, defense systems, financial modeling, medical imaging, inference engines, and real-time decision-making frameworks. It also finds application when classical exact methods become impractical, such as for extremely large sequences, provided the approximation is acceptable.

The method, referred to as FALC (*Fast Approximate Linear Convolution*), is examined here for simplicity on equal-length binary sequences (without loss of generality, since arbitrary numerical sequences can be represented in binary form), while the extension to broader classes of input sequences is also outlined.

The algorithm is **deterministic, single-pass, unconditional** and **parallelizable**, with **tunable accuracy**, operating in $\Theta(n)$ time with $\Theta(1)$ auxiliary memory, and scalability limited only by the memory required to store the I/O sequences.

Within a unified computational framework of generalized interaction operators, the method encompasses cross-correlation and direct convolution and, under suitable assumptions, induces a class of inferential estimators with well-defined asymptotic properties.

A theoretical analysis is developed under a dual paradigm: descriptive and inferential, where the induced operator is studied both as a geometric approximation and as a statistical estimator, and corresponding residual and stochastic error are analyzed to establish asymptotic unbiasedness and consistency under model assumptions.

We carry out extensive empirical benchmarks using "naive" and spectral exact algorithms as reference methods to measure performance and approximation errors.

Paper Organization

This work introduces a novel, deterministic, single-pass, and parallelizable algorithm for **approximating** discrete convolutions. From the algorithm, we define the corresponding FALC (Fast Approximate Linear Convolution) operator, a mathematical object capturing the structure of the convolution. When this operator is employed under suitable probabilistic assumptions, it naturally defines a functional **statistical estimator**.

Our perspective distinguishes clearly between three levels of abstraction: the **algorithm** (computational procedure), the **operator** (mathematical object derived from the algorithm), and the **statistical estimator** (emerging under inferential assumptions).

From a theoretical standpoint, we examine the operator in two complementary ways: a **descriptive** framework, treating it as a geometric **approximation**, and a *distribution-free* **inferential** framework, which establishes asymptotic correctness and consistency of the operator as a statistical estimator.

Empirically, we benchmark the algorithm **relative to** *exact* convolution methods, including the naive $\mathcal{O}(n^2)$ approach and spectral $\mathcal{O}(n \log n)$ methods (e.g., FFT and NTT). The purpose is not to compete, but to provide a reference for assessing error, computational efficiency, and estimation accuracy, particularly for large-scale datasets where exact methods may be impractical.

The manuscript is organized as follows:

- **Part I: Introduction.** We introduce the algorithm, describe the operator derived from it, and clarify the hierarchical relationship with potential statistical estimators, outlining the main methodological features.
- **Part II: Empirical Results.** We benchmark the algorithm **relative to** standard exact methods, quantifying **performance**, **accuracy**, and **stability** across four experimental setups.
- **Part III: Theoretical Framework and Algorithm Derivation.** We present a measure-theoretic foundation for both the geometric-descriptive and inferential perspectives. This section also formalizes a **generalized interaction operator** and **generalized interaction coefficient**, encompassing canonical cross-correlation and direct convolution operators.
- **Part IV: Error Analysis.** The approximation error is analyzed in parallel within both the descriptive and inferential frameworks. We present two sequences of theorems establishing bounds, decay rates, and convergence properties for both deterministic residuals and stochastic errors. Distribution-free optimality results—including consistency, asymptotic unbiasedness, and least-squares minimization—are established in the inferential framework. We also investigate the persistence of accuracy under reduced exact-information guidance. In particular, we study the effect of limiting the number of *waypoints*, i.e., intermediate indices where the exact convolution is computed to guide the approximation. Stability is maintained for constant $W_0 \in \Theta(1)$, independent of the input length n.
- **Applications and Conclusion.** We discuss practical use cases and outline directions for future research.

Part I

Introduction

1 The Discrete Convolution Operator.

1.1 Universality.

Discrete convolution is a fundamental operator across a large range of disciplines. It constitutes the core of fundamental operations; for instance, the classical multiplication algorithm is, in essence, a direct convolution followed by a carry-propagation stage. Moreover, convolution underlies polynomial multiplication, the distribution of sums or differences of random variables, filtering, signal processing, inference engines, machine learning, and high-throughput genomics.

The universality of this operator, spanning applications from radar systems and medical imaging to genomics and convolutional neural networks (CNNs), together with the rapid growth of increasingly demanding computational tasks on massive datasets, has led to the emergence of a fundamental computational limitation: as input lengths n increase, the $\mathcal{O}(n^2)$ complexity of exact computations quickly becomes prohibitive. Spectral methods reduce this complexity to $\mathcal{O}(n \log n)$.

1.2 Intuitive aspects.

The convolution operator can be regarded from different perspectives, each highlighting a different aspect:

- **Sliding Window:** The operator can be seen as a kernel sliding across the input sequence, with each output element representing a weighted sum over the locally overlapping region.
- **Weighted Overlay:** In a complementary, more static view, it can be interpreted as a **weighted overlay** of shifted layers, emphasizing the global structural composition rather than local translations.
- **Diagonal Summations Within the Outer Product:** The operator applied to two input sequences $\mathbf{A}, \mathbf{B}$ appears as sums along specific diagonals of the outer product $\mathbf{A}\mathbf{B}^\top$, where cross-correlation and direct convolution correspond to summations along opposite diagonal orientations, respectively.
- **Mass and Energy Interpretation:** In some contexts, the convolution operator formalizes interacting "masses" where the sum of pairwise products along the convolution captures the local exchange of "energy" between inputs. This interpretation connects directly to physical concepts familiar in mechanics and signal processing, such as moments, energy conservation, and the distribution of influence across the domain.
- **Classical ("Naive") Multiplication:** The classical multiplication algorithm is a direct convolution of digit sequences ($\mathbf{A} * \mathbf{B}$), followed by a $\Theta(n)$ carry propagation to produce the final numeric representation in the desired base (the elementary "shift-and-add" method from

basic arithmetic, where each multiplier digit scales and aligns the multiplicand, and partial sums are accumulated with proper carry propagation). This highlights that discrete convolution is the fundamental operation underlying basic arithmetic, independent of more advanced optimizations such as Karatsuba or FFT-based multiplication, where $(\mathbf{A} * \mathbf{B})$ denotes the direct discrete linear convolution. Base 2 example:

$$
\begin{aligned}
&\mathbf{prod} \leftarrow \mathbf{0}, \quad \text{carry} \leftarrow 0 \\
&\left.\begin{aligned}
&\text{sum} \leftarrow (\mathbf{A} * \mathbf{B})[k] + \text{carry} \\
&\mathbf{prod}[k] \leftarrow \text{sum AND } 1 \\
&\text{carry} \leftarrow \text{sum} \gg 1
\end{aligned}\right\rceil_{k=0}^{2n-2} \\
&\mathbf{prod}[2n-1] \leftarrow \text{carry}
\end{aligned}
$$

- **Polynomial Multiplication and Ring Mapping:** Convolution of two sequences is algebraically equivalent to polynomial multiplication, where each sequence represents the coefficients of a polynomial. This formulation captures all pairwise products of input elements. According to the Discrete Convolution Theorem (also called the Discrete Fourier Transform Convolution Theorem, see Oppenheim & Schafer, 1999), the linear convolution of two sequences can be computed via pointwise multiplication of their discrete Fourier transforms, followed by an inverse transform. Using the Fast Fourier Transform (FFT) for the DFT evaluation, this yields computational complexity $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$, enabling exact computation for large sequences.

- **Distribution of Sums, Differences, and Products of Random Variables:** The convolution operator is used to compute the distribution of sums, differences, and other fundamental operations with random variables, playing a crucial role in probability theory, stochastic modeling, statistics, signal processing, and stochastic systems.

- **Projection and Alignment:** In a Hilbert space, input sequences can be represented as vectors. In such a case the convolution coefficients are given by the inner products of one vector with discrete translations ("shifts") of the other, i.e., projections onto the subspace generated by the shifted vectors. This perspective highlights the operator as a linear mapping defined by a sequence of shift projections.

- **Correlation and Similarity:** Since the cosine between vectors (after centering and normalization) equals the **Pearson correlation coefficient**, convolution operators quantify structural affinity between input sequences. Specifically, as we will see in the following, while **cross-correlation** corresponds to a "parallel" index mapping and the standard Pearson correlation, **direct convolution** represents conceptually an "anti-correlation", arising from the reversal of one input sequence. This interpretative layer can unify geometric and statistical perspectives. (Here, terms such as "anti-correlation" or "contravariance" will be used to denote a **reversal** of

an input sequence; obviously, they are distinct from negative statistical correlation, and are referred to reversed indexing.)

While these intuitive perspectives are not all explicitly used in our study, they provide a rich ensemble of intuitive ideas for our proposed methodology and inform both the design of the operator and subsequent strategies to reduce computational complexity.

1.3 Computational Complexity of the Convolution Operator.

The computation of the discrete convolution according to its definition (irrelevant whether **cross-correlation** or **direct** convolution), which we will refer to as "naive", is bounded by $\mathcal{O}(n^2)$ operations, as each of the $2n-1$ output coefficients $c[k]$, $k = 0, \ldots, 2n-2$, requires a sum over $\mathcal{O}(n)$ **distinct** pairwise products.

More efficient algorithms exploit the **isomorphism between discrete convolution and polynomial multiplication**. Specifically, by employing the Fast Fourier Transform (FFT) to evaluate the input polynomials at suitable points, the overall procedure requires $\mathcal{O}(n \log n)$ operations. Similar results hold over finite fields using the Number Theoretic Transform (NTT), provided that a suitable modulus exists and appropriate primitive roots of unity are available.

Although spectral methods achieve $\mathcal{O}(n \log n)$ complexity, in practice, they can introduce significant overhead due to implementation and architectural factors. Input sequences must be zero-padded to avoid circular convolution, followed by two forward transforms, pointwise multiplication, and an inverse transform. In the case of the NTT, the choice of modulus constrains the admissible input range unless multimodulus techniques based on the Chinese Remainder Theorem (CRT) are employed, which introduces additional computational overhead, both in terms of computation and implementation.

These methods also encounter architectural limitations in several contexts. In streaming or low-latency systems, the global nature of the transform can increase latency, as the entire data block must be accessed before computation; in memory-constrained environments, the additional working storage required for padding can be prohibitive. Moreover, because of "**spectral locality**" (i.e., each spectral coefficient depends on the entire input sequence), even small changes require recomputing the full transform. Conversely, in real-time filtering or inferential engines, computing every coefficient exactly is often unnecessary and may even degrade the quality of the results due to sensitivity to noise or outliers.

1.4 Scientific Heritage

We have mentioned the $\mathcal{O}(n^2)$ complexity bound for the "naive" computation and that spectral methods, most importantly FFT and NTT, achieve substantial $\mathcal{O}(n \log n)$ speedups via a change of representation (Cooley and Tukey, 1965). Nevertheless, they fundamentally remain aggregations of pairwise multiplicative interactions.

Departing from this traditional approach, we conceptualize sequences of length n as "rigid" vectors in Hilbert space, meaning that their internal structural and affine relationships are preserved under orbital propagation throughout the computation and can be exploited. From this viewpoint, it

becomes feasible to reduce computational complexity by approximating correlations (or, in general, interactions) or displacement relationships, rather than evaluating all pairwise interactions explicitly.

The present framework builds upon the classical quantization noise model of Widrow (1956, 1961), later systematized by Widrow and Kollár (2008). Under suitable conditions, quantization errors can be modeled as approximately zero-mean stochastic processes, enabling tractable analysis of fluctuations introduced by discrete representations. Consequently, the exploitation of the interaction manifold rigidity, in both geometrical and statistical terms, allows the design of linear-time approximation schemes that preserve statistical consistency while reducing computational cost.

To preserve structural coherence during estimation, we also incorporate periodic manifold "synchronization" points (i.e., indices where the approximation is "guided" to be exact), reflecting stability principles commonly studied in discrete dynamical systems (Devaney, 1989). Within this general framework, our methodology is informed by several inspiring theoretical perspectives, including quantum algorithmic speedups (Shor, 1994); probabilistic concentration phenomena (Tao, 2012); computational complexity limits in quantum and classical models (Aaronson, 2013); classical fast multiplication algorithms (Schönhage and Strassen, 1971); and sparse recovery guarantees in compressed sensing (Candès, Romberg, and Tao, 2006; Donoho, 2006). These foundations collectively motivate the design of our estimator and support its theoretical and empirical efficiency.

1.5 Cross-correlation and Direct Convolution Operators

In this work, we will adopt a possibly unexpected take on convolution, considering it from a **geometric and statistical** perspective, rather than an algebraic one. To this end, we will introduce in the next sections a general framework that, first of all, will allow seamless integration with **statistical inference**, and also provide the foundation for a **generalized interaction operator** that contains, as particular cases, the following canonical operators

Let $\mathbf{A} := (a[0], \ldots, a[n-1])$ and $\mathbf{B} := (b[0], \ldots, b[n-1])$ be two input sequences of equal length n. We define the index bounds

$$i_{\min}(k) := \max(0, k-(n-1)), \qquad i_{\max}(k) := \min(k, n-1), \tag{1.1}$$

which characterize the following operators:

- **Direct Convolution Operator.** The direct convolution operator is defined as

$$c_*[k] := (\mathbf{A} * \mathbf{B})[k] := \sum_{\substack{0 \le i,j \le n-1 \\ (i+j)=k}} a[i]\, b[j] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[k-i] \tag{1.2}$$

  (Note that the direct convolution operator is denoted by an asterisk $*$, and cross-correlation is denoted by a star $\star$.)

- **Cross-correlation Convolution Operator.** For the cross-correlation, we have

$$c_\star[k] := (\mathbf{A} \star \mathbf{B})[k] := \sum_{\substack{0 \le i,j \le n-1 \\ (j-i)=(n-1)-k}} a[i]\, b[j] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[(n-1)-(k-i)] \tag{1.3}$$

with indices $k \in \{0, \ldots, 2n-2\}$ and output length

$$m := 2n - 1. \tag{1.4}$$

Observe that the reversal of $\mathbf{B}$ in these identities is not an additional operation applied after computing the convolution or cross-correlation. Rather, it is already implicitly contained in the index alignment of the cross-correlation operator. In other words, $(\mathbf{A} \star \mathbf{B})$ can be obtained from $(\mathbf{A} * \mathbf{B})$ simply by reversing the input sequence $\mathbf{B}$, and vice versa. This shows that the two operators are structurally equivalent, with the reverse encoding the natural mapping of indices between them.

The term "cross-correlation convolution operator" is chosen to parallel enquotedirect convolution operator. As will be formalized later, both operators can be seen as instances of a **generalized interaction operator** and a **generalized interaction coefficient**, providing a unified treatment of discrete interactions. This terminology emphasizes the structural similarity of the two operators and foreshadows their unification under the **Generalized Interaction Operator (GIO)** introduced later in this work.

This terminology also offers the advantage, in our setup, of avoiding confusion with classical statistical correlation. In fact, the cross-correlation operator aligns indices in the same way as the **Pearson correlation coefficient**, while the direct convolution can be seen to involve the same computations but with one (and only one) input labeled in reversed order, which can be evocatively interpreted as using a **"contravariance"** instead of a **covariance**, providing an intuitive link for a unified statistical interpretation.

The two operations are in fact linked, as follows. Let $\tilde{\mathbf{B}} := (b[n-1], b[n-2], \ldots, b[0])$ denote the reversed sequence of $\mathbf{B}$. Then, for every k, the following identities hold:

$$c_\star[k] = (\mathbf{A} * \tilde{\mathbf{B}})[k], \qquad c_*[k] = (\mathbf{A} \star \tilde{\mathbf{B}})[k] \tag{1.5}$$

Consequently, any algorithm computing one of the convolutions can produce the output of either operator simply by reversing a single input sequence. (Reversing both would yield a mirrored version of the original operator.) This perspective highlights that, in essence, the two operations are structurally equivalent, emphasizing their unifiable nature rather than treating them as fundamentally distinct.

Remark (Constant-input degeneracy). A particularly revealing limiting case arises when one of the input sequences is a constant vector, denoted as $\mathbf{B} = \{b, b, \ldots, b\}$. In this case, the distinction between direct convolution and cross-correlation collapses into a purely positional symmetry. Specifically, the convolution at index k corresponds to the cross-correlation at the mirrored index $(n-1) - k$. To clarify this, we will explicitly show how the two operations are related when one sequence is constant.

First, recall the definitions of the two operators:

- **Direct Convolution:**

$$c_*[k] := (\mathbf{A} * \mathbf{B})[k] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[k-i]$$

- **Cross-correlation:**

$$c_\star[k] := (\mathbf{A} \star \mathbf{B})[k] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[(n-1) - (k-i)]$$

Now, substitute $\mathbf{B} = \{b, b, \ldots, b\}$ into both operations:

- For direct convolution, we have

$$c_*[k] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[k-i] = b \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]$$

  because b is a constant. This expression simply scales the sum of the elements of $\mathbf{A}$ over the sliding window.

- For cross-correlation, we have

$$c_\star[k] = \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]\, b[(n-1)-(k-i)] = b \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i]$$

  for the same reason: the term b is constant.

However, note that in the cross-correlation expression, the index $(n-1)-(k-i)$ simply reflects the position of b in the reverse order. Therefore, we can write:

$$c_\star[k] = b \sum_{i=i_{\min}(k)}^{i_{\max}(k)} a[i] = \left(\widetilde{\mathbf{c}_*}\right)[k].$$

Thus, we conclude that when $\mathbf{B}$ is a constant vector, the convolution at index k is exactly the same as the cross-correlation at the mirrored index $(n-1)-k$. This positional symmetry indicates that:

$$(\mathbf{A} * \{b, \ldots, b\}) = (\mathbf{A} \star \widetilde{\{b, \ldots, b\}}),$$

where the tilde denotes the reversal of the sequence.

This shows explicitly that when one sequence is constant, the cross-correlation operator is simply the convolution operator applied to the reversed constant sequence.
In this regime, since every element of the kernel is the same constant b, both operators reduce to a scaled summation of the entries of $\mathbf{A}$ over the sliding window. As demonstrated in the previous section, the key observation is that the convolution and cross-correlation operations become identical under these conditions, except for a positional reversal. Specifically, the convolution at index k corresponds exactly to the cross-correlation at the mirrored index $(n-1)-k$.

This cumulative aggregation induces the characteristic piecewise prefix–suffix structure observed in our numerical tests. The positional symmetry between the two operators, confirmed by the constant-input degeneracy, supports their unified treatment as oriented instances of a single **generalized interaction operator** (GIO). This symmetry emphasizes that, under constant-input conditions, the two operations are mirror images of each other, and their relationship can be fully captured within the GIO framework.

While, for clarity, we present a version for equal-length binary sequences, this does not constitute a genuine restriction of generality, since arbitrary numerical sequences admit binary representations and can always be reduced, if necessary, to equal-length forms through padding. The formalization extends *mutatis mutandis* to sequences of different lengths and to general non-binary alphabets, while the binary setting remains particularly natural for efficient hardware implementations.

2 The FALC - Fast Approximate Convolution Algorithm

We introduce the new algorithm, referred to as **FALC** (***F**ast **A**pproximate **L**inear **C**onvolution*), which, by extension, gives its name to the corresponding convolution operator. Our approach is bottom-up: we start from an algorithmic procedure that works efficiently in practice, and then derive the mathematical operator and study its theoretical properties. When used under suitable probabilistic assumptions, this operator naturally defines a functional **statistical estimator**.

For clarity, in this section, we consider binary sequences of equal length, although the extension to arbitrary supports is straightforward. The preliminary aspects highlighted here will support the understanding of the subsequent empirical study and the more detailed theoretical analysis.

On the notion of "approximation". Since the term "approximate" in algorithms can be understood in various ways, we clarify that here it does not refer to deterministic iterative convergence within algorithmic steps for a fixed input (as in, for example, Newton-Raphson or Gauss-Seidel methods). Rather, the approximation—still deterministic—is inherent and of a *structural and, in this case, also statistical nature*: the algorithm produces outputs that approximate the convolution geometrically, with accuracy emerging both in a geometric/descriptive sense and, in this case, also in an asymptotic sense as the input size grows. Under suitable inferential assumptions, this naturally induces a statistical estimator. (Similarly, we are not referring to "numerical" approximations, which are in any case present in all algorithm implementations due to finite-precision arithmetic, rounding, type casting, and similar effects.)

2.1 Waypoints

A defining characteristic of the FALC architecture is the presence of a mechanism that enables a **tunable trade-off between computational performance and accuracy**. This mechanism essentially consists of a set of specific indices at which the convolution is evaluated **exactly**, serving as "guidance" points for the estimation process.

By design, the algorithm always evaluates **exactly** the convolution at three indices: the **central index** $(n-1)$ and the two **convolution boundaries** 0 and $2n-2$. In addition to these three fixed points, the user can specify a set of "intermediate" indices where the convolution is computed exactly. These indices are referred to as ***waypoints*** and are defined independently on the two sides of the central index: toward 0 on the left and toward $2n-2$ on the right. The number of waypoints per side is denoted by W_0. Within the algorithmic specifications, it is an imperative requirement that W_0 remains **constant and independent of the input length** n, that is,

$$W_0 \in \Theta(1).$$

(Clearly, if one allowed W_0 to scale with input length, one would generate a $\mathcal{O}(n^2)$ computation less efficient than the "naive" algorithm.)

For instance, in the empirical study presented in this work, we consider the values $W_0 = 3$ and $W_0 = 0$ (while sequence lengths may range up to 10^8).

2.2 Strategic Placement of Waypoints and Incorporation of Prior Information

The adaptive design of waypoints can optimize the **performance-to-accuracy trade-off** by allocating exact evaluations where the interaction manifold exhibits structures of particular interest. In our empirical study, for simplicity, we place the waypoints equidistantly. In practical applications, the placement of waypoints can be informed by **prior knowledge** about the interaction structure. For instance, they could be strategically placed in **regions exhibiting large local gradients or high spectral density**.

Moreover, from an inferential perspective, placing the waypoints can be interpreted as defining a **prior distribution** over the convolution support that encodes the **available structural information**:

$$P(c \mid \mathbf{A}, \mathbf{B}) \propto P(c)\, P(\mathbf{A}, \mathbf{B} \mid c).$$

In this way, the FALC operator functions as a **Bayesian estimator**, incorporating both prior information and empirical likelihood. When no structural knowledge regarding the interaction curvature is available, a reasonable choice is a **non-informative prior**. Under this interpretation, the uniform placement example can be viewed, following Jeffreys' framework, as defining priors that do not favor any region of the parameter space in the absence of information.

It is important to note that the equispaced configuration serves only as a **baseline design** and for simplicity. More generally, the waypoint distribution should incorporate any available structural information regarding the interaction. For example, when empirical or spectral diagnostics provide insights into the local variability of the interaction, the waypoint prior should be adapted by concentrating synchronization nodes in regions with stronger gradients, higher curvature, or increased spectral density. Such informed placement may improve the accuracy of the approximation by allocating exact evaluations in regions of greatest structural complexity.

2.3 FALC's Computational Complexity and Features

Key points regarding FALC's computational efficiency, which will be demonstrated following the empirical study, are:

- **Linear computational cost:** The algorithm complexity is $\Theta(n)$, as it deterministically performs a single, unconditional, linear pass over the input sequences of length n to produce the output of length $2n - 1$.
- **Constant auxiliary space:** Memory usage **beyond the input and output** arrays is $\Theta(1)$.

2.4 Properties of the Algorithm

- **Deterministic:** The algorithm does not invoke any random number generator (RNG) and produces outputs fully determined by the input.
- **Single-Pass:** Only one linear (possibly in parallel) scan of the input is required, making it suitable for streaming and memory-limited environments.

- **Unconditional:** Meaning that the algorithm contains no IF-like statements or any form of branching, ensuring linear, data-driven computation.
- **Parallelizable:** The computation can be split into independent lattice segments, allowing it to run across multiple cores or GPU threads. This parallel execution preserves $\Theta(n)$ complexity with minimal synchronization overhead and can improve execution time.
- **Approximate with Tunable Accuracy:** The algorithm provides an approximation of the coefficients, and waypoints can be placed strategically to improve accuracy, enabling flexible adaptation to the underlying structure or specific application requirements.
- **Scalable:** Memory usage is $\Theta(n)$ with strictly constant auxiliary space $\Theta(1)$, and no intrinsic algorithmic constraints, allowing operation on sequences of extreme length, limited only by available physical memory.

2.5 Extension to General Sequences

While in our empirical study, we use binary sequences for simplicity, the algorithm can be extended to other data types. Binary sequences also represent a challenging scenario due to discrete jumps and possibly minimal local correlation. For general sequences in $\ell^2(\mathbb{Z})$, the propagation logic remains unchanged. Smoothness or oversampling in the input typically improves approximation accuracy relative to the binary baseline. Thus, binary tests may provide a conservative estimate of performance for broader signal classes.

2.6 Stability and Convergence

The reduction from $\mathcal{O}(n^2)$ to $\Theta(n)$ complexity in our algorithm is achieved by formulating the convolution as the propagation of an interaction state, rather than as a sequence of independent inner products. Since adjacent interactions share overlapping elements, the update of each interaction coefficient depends strictly on local differences. By initializing the state at a maximal-overlap index to optimize the signal-to-noise ratio and minimize variance, the algorithm updates the interaction state incrementally. Each subsequent coefficient is obtained in constant $\Theta(1)$ time, ensuring that the output waveform—defined as the **ordered sequence of convolution coefficients**—is recovered through state propagation rather than exhaustive pairwise products.

The statistical properties of the induced estimator $\widehat{g}[d]$ satisfy the following convergence criteria:

- **Asymptotic Consistency:** For stochastic sequences, the estimator is consistent in probability. Specifically, the local residual $\varepsilon[d] := \widehat{g}[d] - g[d]$ satisfies $\varepsilon[d] = \mathcal{O}_p(n^{-1/2})$, ensuring that the per-element error vanishes as the lattice density increases. For deterministic sequences under (A1), the error remains uniformly bounded by the structural curvature.
- **Integral Preservation:** The global sum of interaction coefficients is preserved up to a cumulative deviation $\mathcal{E}_n = \mathcal{O}_p(\sqrt{n})$. This scaling is consistent with a stable behavior of the aggregate interaction "mass" in the observed regimes, as local fluctuations appear to be averaged out at the global level by the sequential aggregation structure of the algorithm. A more detailed

probabilistic interpretation of this effect, specific to the algorithmic model introduced later in the paper, will be provided in the modeling section.

- **Bounded Residual Drift:** The cumulative error process along a radial branch follows a square-root scaling, $|\mathcal{E}_h| = \mathcal{O}_p(\sqrt{h})$. This prevents deterministic drift and ensures that the global waveform structure is preserved, maintaining a relative error of order $\mathcal{O}_p(n^{-1/2})$ across the entire manifold.

Empirical results confirm that the relative estimation error scales as $\mathcal{O}_p(n^{-1/2})$, while the per-element computational cost remains strictly $\Theta(1)$.

3 Levels of Analysis

The algorithm is analyzed both **empirically** and **theoretically**, providing complementary perspectives:

- **Geometric Analysis:** Treats the output as an approximation of the convolution manifold. Residuals represent signed deviations, focusing on the geometric and topological structure of the interaction.
- **Statistical Analysis:** Treats residuals as realizations of a stochastic error process. Within a distribution-free framework, the algorithm is interpreted as an estimator, allowing assessment of consistency and asymptotic unbiasedness.

Empirical evaluation demonstrates practical performance, while theoretical analysis confirms asymptotic properties under both deterministic and stochastic interpretations.

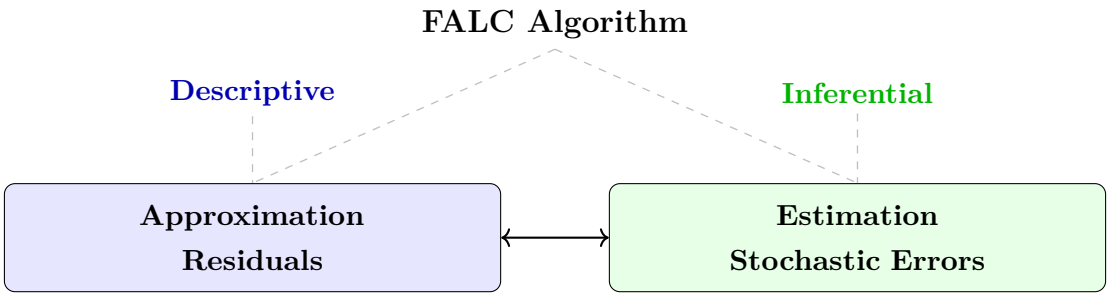

Figure 1: **Duality of our theoretical approach.**

TECHNICAL DISCLOSURE NOTICE

This document contains a deliberately limited excerpt from a broader technical corpus associated with the **G-Bot Algorithmic research architecture**.

The full system includes multiple modular components, including a satellite research project focused on **high-performance convolution estimation in** $\Theta(n)/\Theta(1)$ aux space.

The complete specification exceeds **100 pages** and is distributed exclusively as part of the licensed system package.

MODULE EXCERPT — CONTINUATION REDACTED

This section represents a partial disclosure of a modular subsystem. The complete convolution framework continues beyond this point and is not publicly released.

[OMISSIS — FULL CONVOLUTION FRAMEWORK EXCEEDS 100 PAGES]

Applications and Conclusion

Discrete convolution is a foundational operation across scientific computing, engineering systems, and large-scale data processing architectures. As such, its computational complexity directly propagates into system-level constraints in latency, energy consumption, and deployable model scale.

Classical spectral methods, including FFT and NTT-based approaches, operate at $\mathcal{O}(n \log n)$ complexity and typically require full-sequence availability, zero-padding, and auxiliary workspace proportional to input size. While exact, these constraints introduce structural limitations in memory-bound or streaming environments, particularly in real-time and edge-deployed systems.

In contrast, FALC operates in $\Theta(n)$ time with $\Theta(1)$ auxiliary memory overhead beyond input and output buffers. Computation is strictly sequential, with no requirement for global transforms, intermediate state expansion, or external workspace allocation.

This shifts the computational model from *transform-based batch processing* to *streaming-native execution*, where convolution is evaluated incrementally as data arrives.

The method is therefore applicable in regimes where exact coefficient recovery is not the primary objective, but where structural fidelity of interactions is sufficient or preferred. This includes statistical

inference pipelines, signal processing systems, large-scale pattern recognition, and exploratory high-throughput analysis.

Across these domains, the critical constraint is not mathematical exactness, but system throughput under bounded compute and memory budgets. In this setting, controlled approximation with explicit error structure becomes a design feature rather than a limitation.

The proposed formulation avoids global transforms entirely, eliminating both the algebraic constraints associated with number-theoretic transforms and the memory overhead of spectral methods. Its linear structure enables processing of sequences at scales that are typically inaccessible to exact convolution pipelines, subject only to I/O proportional scaling.

Empirical evaluation indicates stable performance on sequences exceeding $n \sim 3\times10^8$ on commodity high-memory systems, whereas spectral methods encounter practical limits at significantly lower scales due to memory and transform overhead.

Residual behavior remains consistent with theoretical bounds under tested regimes, confirming predictable error propagation in applied settings.

Overall, the combination of linear-time complexity and constant auxiliary memory positions FALC as a viable computational primitive for high-throughput systems where latency, scale, and resource constraints dominate architectural design.

Representative Applications

- **Defense, Aerospace, and Remote Sensing:** Radar, sonar, LiDAR, satellite imaging, synthetic aperture radar (SAR), interferometric reconstruction, distributed sensor fusion, pulse compression, and high-frequency target tracking.

  In these environments, convolution and cross-correlation are core primitives embedded in detection, classification, and situational awareness pipelines. The dominant constraint is not model design, but real-time execution under strict latency, power, and compute limitations across distributed and often edge-deployed systems (airborne, orbital, and autonomous platforms).

  FALC enables a structural shift from batch-oriented signal reconstruction to continuous-time inference by reducing convolutional complexity while preserving controllable approximation bounds. This directly impacts mission-critical responsiveness in contested or bandwidth-limited environments.

- **Computational Biology and Life Sciences:** Genome-wide similarity scoring, motif discovery, protein–protein interaction networks, structural genomics, pathway inference, and large-scale population genomics.

  Biological data processing at scale is increasingly constrained by pairwise similarity and kernel-based computations over extremely large state spaces. These operations exhibit convolution-like structure over sequence, graph, and spatial domains.

  FALC enables tractable approximation of these operations at scale, allowing near real-time inference over datasets that are otherwise restricted to offline batch processing, while maintaining controlled error propagation in downstream statistical models.

- **Medical Imaging and Diagnostic Systems:** High-resolution MRI, CT, PET, ultrasound reconstruction, multi-modal image fusion, temporal volumetric analysis, and integrated imaging-genomic diagnostic pipelines.

  Modern medical imaging systems rely heavily on iterative reconstruction and convolutional filtering over large volumetric datasets. These pipelines are computationally intensive and often limit real-time diagnostic usage.

  FALC reduces reconstruction latency by replacing exact convolutional kernels with scalable approximations, enabling faster clinical decision support and more responsive multi-modal diagnostic systems.

- **Information Security, Cyber Defense, and Finance:** Real-time anomaly detection, network traffic analysis, intrusion detection systems, fraud detection, high-frequency trading, and cryptanalytic signal processing.

  Across these domains, convolution-like operations appear in correlation estimation, feature extraction, and temporal smoothing of high-dimensional streaming data. The main constraint is throughput under non-stationary, adversarial, or rapidly evolving conditions.

  FALC reduces computational overhead in high-frequency correlation analysis, enabling faster adaptation cycles in environments where latency directly determines informational advantage.

- **Artificial Intelligence and Machine Learning:** Convolutional neural networks, attention mechanisms, transformer architectures, graph neural networks, diffusion models, and large-scale representation learning systems.

  A significant fraction of modern AI compute is dominated by kernel-based operations, including attention matrices and convolution-like aggregation operators. These operations scale poorly in high-dimensional regimes, often becoming the dominant cost in both training and inference.

  FALC enables scalable approximation of these primitives, supporting:

  - reduced inference latency for large-scale models
  - deployment of transformer-class architectures in constrained environments
  - extension of long-range dependency modeling beyond quadratic cost regimes

- **Scientific Simulation and Complex Systems:** Molecular dynamics, lattice field models, fluid dynamics, electromagnetic propagation, and climate and multi-scale physical simulations.

  Many physical systems are governed by integral operators or convolutional kernels acting on discretized state spaces. These dominate runtime in high-resolution or multi-scale simulations.

  FALC reduces the cost of repeated kernel evaluation while preserving global structure, enabling higher-resolution simulation within fixed computational budgets.

- **Signal Processing and Communications:** Audio and visual signal processing, multi-sensor fusion, wireless communication systems, modulation/demodulation pipelines, adaptive filtering, and real-time synthesis.

Convolutional operations underpin filtering, encoding, and reconstruction across nearly all modern communication systems. The limiting factor is often energy and latency rather than algorithmic expressiveness.

FALC enables low-latency, energy-efficient signal processing in noisy, bandwidth-constrained, and time-critical environments.

- **Autonomous Systems and Robotics:** SLAM, multi-sensor fusion, predictive control, trajectory optimization, UAV navigation, autonomous vehicles, industrial robotics, and distributed swarm systems.

  Autonomous systems rely heavily on real-time perception and decision loops where convolutional and correlation operations dominate perception pipelines.

  FALC enables continuous inference loops under embedded constraints, reducing compute load while maintaining stable operational accuracy in dynamic environments.

- **Cloud Computing and High-Performance Infrastructure:** Large-scale tensor operations, GPU/TPU acceleration pipelines, distributed inference systems, and data-center-scale model execution.

  At hyperscale, convolutional and kernel operations represent a significant portion of compute cost and energy consumption. Optimizing these primitives directly translates into infrastructure-level cost reduction.

  FALC reduces runtime complexity and energy usage for large-scale workloads, improving throughput per watt in distributed compute environments.

- **Emerging Computational Domains:** Quantum-inspired classical simulation, high-dimensional statistical inference, secure multi-party computation, geospatial intelligence, swarm coordination, edge AI systems, and large-scale decision support infrastructures.

  These domains are characterized by interaction-rich systems where exact convolution becomes computationally infeasible. Approximate convolution primitives allow tractable modeling of large-scale dependency structures under constrained compute budgets.