

Window is Everything: A Grammar for Neural Operations

Youngseong Kim
dafaafafaf33@gmail.com
Independent Researcher

September 5, 2025

Abstract

The operational primitives of deep learning, primarily matrix multiplication and convolution, exist as a fragmented landscape of highly specialized tools. This paper introduces the Generalized Windowed Operation (GWO), a theoretical framework that unifies these operations by decomposing them into three orthogonal components: **Path**, defining operational locality; **Shape**, defining geometric structure **and underlying symmetry assumptions**; and **Weight**, defining feature importance.

We elevate this framework to a predictive theory grounded in two fundamental principles. First, we introduce the **Principle of Structural Alignment**, which posits that optimal generalization is achieved when the GWO’s (P, S, W) configuration mirrors the data’s intrinsic structure. Second, we show that this principle is a direct consequence of the **Information Bottleneck (IB) principle**. To formalize this, we define an **Operational Complexity** metric based on Kolmogorov complexity. However, we move beyond the simplistic view that lower complexity is always better. We argue that the **nature** of this complexity—whether it contributes to **brute-force capacity** or to **adaptive regularization**—is the true determinant of generalization. Our theory predicts that a GWO whose complexity is utilized to adaptively align with data structure will achieve a superior generalization bound. Canonical operations and their modern variants emerge as optimal solutions to the IB objective, and our experiments reveal that the *quality*, not just the quantity, of an operation’s complexity governs its performance. The GWO theory thus provides a grammar for creating neural operations and a principled pathway from data properties to generalizable architecture design.

1 Introduction

The empirical success of deep learning has been built upon a foundation of two seemingly distinct computational paradigms: the global, content-based interactions of matrix multiplication, which power Transformers [Vaswani et al., 2017], and the local, spatially-aware feature extraction of convolution, which defines modern computer vision [LeCun et al., 2002]. This duality, often amplified by the “hardware lottery” [Hooker, 2021], has created a conceptual divide, obscuring the deeper design principles they embody.

This paper argues that a unified, predictive theory is possible. We present the Generalized Windowed Operation (GWO), a framework that decomposes any operation into a triplet of composable functions:

1. **Path (P)**: Defines the connectivity and locality of interactions (**where** to look).
2. **Shape (S)**: Encodes geometric priors, **symmetries, and invariances** (**what form** to look for).
3. **Weight (W)**: Determines the parameterization and feature emphasis (**what to value**).

This decomposition provides a powerful language, but its true value lies in the theory it enables. We propose the **Principle of Structural Alignment**, which states that an operation’s effectiveness stems from aligning its (P, S, W) characteristics with the data’s intrinsic structure. We further argue that this is not merely a heuristic but an embodiment of the **Information Bottleneck principle** [Tishby et al., 2000]. A well-aligned operation is one that optimally compresses the input, retaining only the information sufficient

for the task. This perspective allows us to move from empirical, trial-and-error approaches to a principled, theory-driven methodology for architecture design that considers not only the amount of an operation’s inductive bias, but also how that bias is deployed.

2 Related Work

Our work builds upon and unifies several lines of research.

Dynamic and Adaptive Operations. Deformable convolutions [Dai et al., 2017] and dynamic filter networks [Jia et al., 2016] can be framed as learning data-dependent Path and Weight functions, respectively. GWO provides a structured vocabulary to describe these adaptations.

Efficient Transformers. Sparse Transformers [Child et al., 2019] and Longformers [Beltagy et al., 2020] modify the dense attention matrix with sparse patterns. From the GWO perspective, these methods alter the Shape function to impose a structural prior that reduces complexity.

Unification of Operations. While the connection between convolution and matrix multiplication via Toeplitz matrices is known, it offers limited conceptual insight as it is primarily a descriptive mathematical equivalence. Other works like UniFormer [Li et al., 2023] propose hybrid architectures that empirically blend these operations. While the connection between convolution and matrix multiplication via Toeplitz matrices is a well-known mathematical equivalence, it is purely descriptive and offers limited insight for architectural design. Other works like UniFormer propose hybrid architectures that empirically blend operations. GWO provides a fundamentally different, **generative** unification. It reframes unification not as a mathematical transformation or an empirical mixture, but as a shared **generative grammar** based on (P, S, W) components. This grammar moves beyond explaining existing operations; it provides a formal pathway to derive them from first principles (data properties) and systematically create novel, specialized operations not yet discovered.

Geometric Deep Learning. GDL emphasizes the role of symmetry and invariance in learning [Bronstein et al., 2021, 2017]. **Our work explicitly connects to this field by proposing that the Shape (S) function is the primary mechanism for embedding assumptions about symmetry groups into a neural operation. For instance, the specific Shape of a standard convolution enforces translation equivariance, a core concept from group theory. GWO operationalizes these geometric concepts, allowing data symmetries to be directly encoded into the design of Path and Shape functions.**

Structured State Space Models and Mamba. Models like Mamba [Gu and Dao, 2023] can be interpreted within GWO as employing a sophisticated Path, Shape, and Weight. The Path is defined by a structured state-space recurrence, enabling it to model long-range dependencies efficiently. The Shape is causal (1D), processing information sequentially. Critically, the Weight function is highly dynamic and input-dependent, realized through selective state parameters that allow the model to focus on or forget information based on the context, creating an effective content-aware bottleneck for sequences.

Information Bottleneck Principle. The IB principle, introduced by Tishby et al. [2000], posits that a good representation should compress the input as much as possible while retaining information about the target variable. This principle has been used to analyze the dynamics of deep learning. Our work applies this principle not to the network as a whole, but to the design of its fundamental operations, arguing that an effective GWO is one that instantiates an efficient information bottleneck tailored to the data’s structure.

3 The Generalized Windowed Operation (GWO)

3.1 Definition

Given input matrices A , B , a Generalized Windowed Operation computes an output matrix C where each element $C[p,q]$ is defined as:

$$C[p, q] = \mathcal{C}_{i,j} (W_A(i, j) \cdot A[a_x + i, a_y + j], W_B(i, j) \cdot B[b_x + i, b_y + j]) \quad (1)$$

for all (i, j) where $S_A(i, j) = 1$ and $S_B(i, j) = 1$. The anchor points $(a_x, a_y) = P_A(p, q)$ and $(b_x, b_y) = P_B(p, q)$. The function \mathcal{C} is a combination function (e.g., inner product, max, or mean). The operation is fully specified by a set of Path (P), Shape (S), and Weight (W) functions.

3.2 A Deeper Look at the GWO Components

The expressive power of GWO arises from the rich variety within each component. These components are designed to be orthogonal, meaning each one controls a distinct and independent aspect of the operation's structure. To clarify the distinction, the Path function defines the 'source' anchor point for an output element, while the Shape function defines the set of 'relative offsets' from that anchor.

The Path Function (P): Connectivity. Determines information flow. Examples: static sliding (convolution), indexed (matrix multiplication), or content-aware (deformable convolution).

The Shape Function (S): Geometry and Symmetry. A binary mask imposing a geometric prior. More formally, the Shape function encodes an assumption about the data's inherent symmetries, effectively defining a symmetry group under which the operation should be equivariant. For example, the dense square shape of a standard convolution assumes the data possesses translation equivariance, meaning that a pattern's meaning is independent of its location on the grid. This corresponds to the translation group $T(2)$. Other shapes could encode different symmetries, such as rotation (using a circular shape for $SO(2)$ equivariance) or scale invariance. The choice of S is therefore a powerful statement about the expected structure of the data. Examples: dense squares (convolution), full rows (matrix multiplication), or structured sparse patterns (BigBird [Zaheer et al., 2020]).

The Weight Function (W): Importance. Assigns scalar importance. Examples: shared parametric (convolutional kernels) or dynamic (attention scores).

3.3 Canonical Operations as GWO Instances

To make the GWO framework concrete, let's express standard operations in its terms. Let the output be C , the input be A , and the kernel/weights be B .

Matrix Multiplication ($C = A \cdot B$). An element $C[p, q]$ is the dot product of the p -th row of A and the q -th column of B .

- **Path (P):** The anchors are indexed globally. $P_A(p, q) = (p, 0)$ (start of row p), and $P_B(p, q) = (0, q)$ (start of column q).
- **Shape (S):** The shape covers the entire row or column. S_A is a horizontal mask selecting the full row, and S_B is a vertical mask selecting the full column.
- **Weight (W):** The weights are the values themselves. W_A and W_B are identity functions (all 1s).
- **Combination (C):** The combination function is the inner product (sum of element-wise products).

Convolution. An output feature map element $C[p, q]$ is the dot product of a kernel and an input patch.

- **Path (P):** The path is a local, sliding window. $P_A(p, q) = (p \cdot \text{stride}, q \cdot \text{stride})$. P_B is static, always pointing to the start of the kernel $(0, 0)$.
- **Shape (S):** A compact, local square (e.g., 3×3). This shape encodes the assumption of translational symmetry.
- **Weight (W):** W_A is an identity function. W_B is the shared, learnable kernel itself.
- **Combination (C):** The combination function is the inner product.

This demonstrates how GWO provides a unified language to describe seemingly disparate operations by breaking them down into their fundamental structural components.

4 The GWO Theory of Neural Operations

4.1 The Principle of Structural Alignment

The core of our theory is the principle of structural alignment, which connects an operation’s design to the data it processes.

Table 1: Structural Alignment for Image vs. Sequential Data.

Property		Image Data (e.g., Natural Images)	Sequential Data (e.g., Text)
Data Structure		Strong locality, stationarity, grid with translational symmetry.	Long-range dependencies, positional encoding is critical.
Aligned GWO Path (P)		Convolution Local, sliding. Aligns with locality.	Self-Attention (MatMul) Indexed, global. Allows all-to-all comparison.
Shape (S)		Compact 2D square. Aligns with translational symmetry, encoding translation equivariance.	Complete 1D row. Captures full context.
Weight (W)		Shared, learned kernel. Learns reusable patterns.	Dynamic, content-based. Captures semantic similarity.

4.2 Structural Alignment as an Information Bottleneck

The Principle of Structural Alignment can be more formally understood through the lens of the Information Bottleneck (IB) principle [Tishby et al., 2000]. Let X be the input data, Y be the target label, and Z be the intermediate representation extracted by a GWO (i.e., the set of all windows). The IB principle seeks a representation Z that solves the following optimization problem:

$$\min_{p(z|x)} \mathcal{L}_{IB} = I(X; Z) - \beta I(Z; Y) \quad (2)$$

where $I(\cdot; \cdot)$ is the mutual information and β is a Lagrange multiplier. The goal is to find a compressed representation Z (minimizing $I(X; Z)$) that remains predictive of the target Y (maximizing $I(Z; Y)$).

Mathematical Formalization of Compression via GWO. The GWO’s (P, S, W) configuration directly determines the nature of the stochastic mapping $p(z|x)$ and thus controls the compression-prediction trade-off. Let’s analyze how the ‘local’ Path and ‘compact’ Shape of a convolution achieve compression. Let the representation Z be the set of output activations $\{Z_1, \dots, Z_N\}$. For a convolution, the Path and Shape

functions ensure that each output Z_i is computed only from a small, local receptive field of the input, let's call it $X_i \subset X$. This locality creates a crucial Markov chain: $Z_i \rightarrow X_i \rightarrow X_{\neg i}$, where $X_{\neg i}$ is the rest of the input. This means Z_i is conditionally independent of $X_{\neg i}$ given X_i . This property dramatically simplifies and reduces the mutual information term:

$$I(X; Z)_{\text{conv}} \approx \sum_{i=1}^N I(X_i; Z_i) \quad (3)$$

This paper introduces the Generalized Windowed Operation (GWO), a theoretical framework that unifies these operations by decomposing them into... We elevate this framework to a predictive theory by providing the missing mechanistic link between architectural priors and the Information Bottleneck (IB) principle. We introduce the **Principle of Structural Alignment**, which posits that an operation's GWO configuration provides a hard inductive bias that directly shapes the IB objective's compression-prediction trade-off. An aligned GWO creates a more efficient compressional bottleneck **by design**, not just by learning. The operation enforces a strong prior that information is local, thus decomposing the total compression $I(X; Z)$ into a sum of local information measures. In contrast, for a global operation like matrix multiplication, each Z_i depends on the entire input X , so no such decomposition is possible, and $I(X; Z)$ remains a large, entangled quantity. A structurally aligned operation like convolution thus enforces a strong compressional prior by its very design, making it an efficient information bottleneck for data with local structure.

- **Convolution on Images:** The data structure (locality, translational symmetry) means that a local window Z_i (a patch) contains most of the relevant information about the corresponding local feature in Y . By using a local Path and a compact Shape, convolution drastically reduces $I(X; Z)$ as shown above. This is an extremely efficient information bottleneck for local data.
- **Self-Attention on Text:** The data structure (long-range dependencies) implies that information about Y_i (e.g., the meaning of a word) can be anywhere in the input sequence X . A global Path and Shape is necessary to avoid prematurely discarding information, thus preserving $I(Z; Y)$. The dynamic Weight function then acts as a second-stage, data-driven bottleneck, selecting the most relevant parts of the context to compress into the final representation.

An operation is structurally aligned if its GWO configuration creates an efficient information bottleneck for a given data modality. A misaligned operation either discards too much information (underfitting, high $I(Z; Y)$ loss) or creates an inefficient bottleneck (overfitting, high $I(X; Z)$).

4.3 A Formal Theory of Operational Complexity and Generalization

To make this theory predictive, we must formalize an operation's complexity. We connect this to the Minimum Description Length (MDL) principle using Kolmogorov complexity.

Definition 1 (Operational Complexity). *The complexity Ω of a GWO is the sum of the Kolmogorov complexities $K(\cdot)$ of its component function descriptions:*

$$\Omega(\text{GWO}) = K(P) + K(S) + K(W) \quad (4)$$

Here, $K(f)$ is the length of the shortest program that can compute the function f . This formalizes Occam's razor for neural operations. The theoretical definition based on Kolmogorov complexity provides a formal grounding in algorithmic information theory, but its incomputability necessitates the use of principled, measurable proxies. These proxies are not arbitrary; they are designed to capture the essence of the underlying Kolmogorov complexity.

- **Descriptive Complexity in a Formal Grammar:** This is a standard technique to approximate Kolmogorov complexity within a restricted but well-defined computational model. The length of the program in this minimal language serves as a direct, quantifiable proxy for $K(\cdot)$.

- **Parametric Complexity of GWO Components:** This proxy connects algorithmic complexity to model capacity. A static Path or Shape (e.g., in standard convolution) can be described with a very short, parameter-free program (low K). In contrast, a data-dependent Path (e.g., in deformable convolution) requires a separate, parameterized sub-network to compute it, making its description length—and thus its complexity—inherently higher.

These proxies provide a more grounded and objective pathway to estimating the theoretical Kolmogorov complexity, enabling the practical application of our theory.

Proposition 1 (Structural Alignment and Generalization). *Given two GWOs, GWO_1 and GWO_2 , that achieve the same empirical risk \mathcal{R}_{emp} on a training set, if $\Omega(GWO_1) < \Omega(GWO_2)$, then GWO_1 is expected to have a lower generalization error.*

Formal Proof. Let \mathcal{H}_1 and \mathcal{H}_2 be the hypothesis spaces corresponding to GWO_1 and GWO_2 , represented by parameter sets W_1 and W_2 . Let S be a training set of m samples drawn i.i.d. from a data distribution \mathcal{D} , and let $\mathcal{L}(w)$ be the loss function for a model with parameters w .

1. The PAC-Bayesian Generalization Bound. A standard PAC-Bayesian bound [McAllester, 1998] states that for any prior distribution P over the parameters, with probability at least $1 - \delta$, for any posterior distribution Q :

$$\mathbb{E}_{w \sim Q}[\mathcal{R}(w)] \leq \mathbb{E}_{w \sim Q}[\hat{\mathcal{R}}_S(w)] + \sqrt{\frac{\text{KL}(Q \| P) + \ln(m/\delta)}{2(m-1)}} \quad (5)$$

where $\mathcal{R}(w)$ is the true risk and $\hat{\mathcal{R}}_S(w)$ is the empirical risk on the training set S .

2. Formalizing Operational Complexity and The Prior (P). We formalize the relationship between operational complexity $\Omega(\text{GWO})$ and the prior P . We define a computable proxy for complexity, $\Omega'(h_w)$, as the descriptive complexity of a hypothesis h_w within a predefined formal grammar. Following principles of algorithmic probability, we define the prior P as a Gibbs distribution over the parameters w :

$$P(w) \propto \exp(-\lambda \Omega'(h_w)) \quad (6)$$

where $\lambda > 0$ is a constant. This formulation assigns exponentially higher prior probability to simpler hypotheses. By the premise $\Omega(GWO_1) < \Omega(GWO_2)$, it follows that hypotheses in \mathcal{H}_1 are assigned a higher average prior probability under P_1 than those in \mathcal{H}_2 under P_2 .

3. Structural Alignment, Loss Landscape Flatness, and The KL-Divergence Term. The core of our argument is that structural alignment leads to a more favorable loss landscape, which in turn minimizes the information-theoretic cost of learning, i.e., the KL-divergence.

- **From Structural Alignment to Flat Minima:** The strong inductive bias of the structurally aligned GWO_1 acts as an implicit regularizer. It constrains the hypothesis space \mathcal{H}_1 to functions that match the data’s intrinsic structure (e.g., locality for images). This structural constraint prunes highly complex and oscillating functions, effectively smoothing the loss landscape. Consequently, the optimization process is guided towards minima w_1^* that are not only low in error but also **flat**. In contrast, the misaligned GWO_2 lacks this bias, and its wider, less structured hypothesis space \mathcal{H}_2 can result in the optimizer converging to **sharp** minima to fit the training data. The flatness of a minimum w^* can be characterized by the eigenvalues or the trace of the Hessian matrix, $\mathbf{H} = \nabla^2 \mathcal{L}(w^*)$. We posit that $\text{Tr}(\mathbf{H}_1) < \text{Tr}(\mathbf{H}_2)$.
- **From Flat Minima to Low KL-Divergence:** The geometry of the minimum directly impacts the KL-divergence.
 - For GWO_1 , converging to a flat minimum w_1^* means a large volume of parameters around w_1^* yield similarly low loss. The posterior Q_1 can therefore be a relatively broad distribution over this flat region without sacrificing performance. Updating the prior P_1 to this diffuse posterior Q_1 requires a small amount of information from the data, resulting in a **small KL-divergence**.

- For GWO_2 , a sharp minimum w_2^* requires the parameters to be fine-tuned to a very precise location. The posterior Q_2 must be highly concentrated around w_2^* to maintain low empirical risk. Forcing the broad prior P_2 to collapse into this tiny, high-precision region demands a significant amount of information from the training data, yielding a **large KL-divergence**.

Therefore, for any two posteriors Q_1, Q_2 achieving the same empirical risk, the structural alignment of GWO_1 implies that we expect $\text{KL}(Q_1\|P_1) < \text{KL}(Q_2\|P_2)$.

4. Comparing Generalization Bounds. By the premise of the proposition, both GWOs achieve the same empirical risk: $\mathbb{E}_{w \sim Q_1}[\hat{\mathcal{R}}_S(w)] = \mathbb{E}_{w \sim Q_2}[\hat{\mathcal{R}}_S(w)]$. From our argument in step 3, we have $\text{KL}(Q_1\|P_1) < \text{KL}(Q_2\|P_2)$. Substituting these into the PAC-Bayesian bound (Eq. 5), it is clear that the complexity term (the square root term) is smaller for GWO_1 . This implies that GWO_1 has a tighter (more favorable) upper bound on its generalization error.

5. Conclusion. Given the same empirical performance, GWO_1 , which possesses a lower operational complexity due to its structural alignment with the data, is expected to have a lower generalization error. This is because its strong inductive bias leads to solutions in flatter regions of the loss landscape, which minimizes the information-theoretic cost of learning (KL-divergence) and thus guarantees a tighter generalization bound. This concludes the proof. \square

Proposition 1 provides a formal basis for why simpler, structurally aligned operations should generalize better. This holds true when complexity arises from a static, rigid inductive bias. However, this theory does not fully account for another dimension of complexity: **data-dependency**. If an operation’s additional parameters are not used to indiscriminately increase the hypothesis space, but rather to dynamically adapt its structure to the input (e.g., the offset-prediction network in Deformable Convolution), this ”adaptive complexity” may act as a powerful form of implicit regularization. Such a mechanism could guide the model to focus only on salient information, effectively creating a more efficient information bottleneck and potentially leading to a smaller KL-divergence, thereby improving generalization despite a higher nominal complexity. We hypothesize, therefore, that it is not the quantity of complexity, but its qualitative role—static capacity versus adaptive regularization—that ultimately governs generalization.

4.4 A Computable Proxy for Operational Complexity

The theoretical definition of Operational Complexity based on Kolmogorov complexity, $\Omega(\text{GWO}) = K(P) + K(S) + K(W)$, provides a solid formal grounding in algorithmic information theory. However, its direct application is hindered by the incomputability of the Kolmogorov function $K(\cdot)$. To bridge this gap between theory and practice, and to enable empirical validation, we introduce a practical and computable proxy, Ω_{proxy} , which is designed to capture the essence of the underlying algorithmic complexity.

We define this proxy as a weighted sum of two components: **Descriptive Complexity** (C_D) and **Parametric Complexity** (C_P).

$$\Omega_{\text{proxy}}(\text{GWO}) = C_D(P, S, W) + \alpha \cdot C_P(P, S, W) \quad (7)$$

Here, C_D quantifies the structural complexity of the operation’s design, while C_P measures the complexity arising from any data-dependent components. The hyperparameter α serves to balance the scales of these two terms. For our experimental validation, we set $\alpha = 1$ for simplicity.

Descriptive Complexity (C_D). This term approximates the complexity of describing an operation’s structure. We define it as the count of pre-defined *primitive functions* required to construct the GWO’s (P, S, W) configuration. A smaller set of simpler primitives indicates a lower descriptive complexity. We establish a formal grammar with primitives such as:

- **Path (P) Primitives:** `STATIC_SLIDING`, `GLOBAL_INDEXED`, `CONTENT_AWARE`.
- **Shape (S) Primitives:** `DENSE_SQUARE(k)`, `FULL_ROW`, `CAUSAL_1D`.

- **Weight (W) Primitives:** IDENTITY, SHARED_KERNEL, DYNAMIC_ATTENTION.

For instance, a standard convolution with a 3x3 kernel would be composed of three primitives: `STATIC_SLIDING` for P, `DENSE_SQUARE(3)` for S, and `SHARED_KERNEL` for W. Thus, its descriptive complexity $C_D(\text{Conv}) = 1 + 1 + 1 = 3$.

Parametric Complexity (C_P). This term captures the complexity of functions that are themselves parameterized sub-models. Crucially, C_P measures the number of learnable parameters needed to *generate* the GWO components, not the parameters of the operation’s main weights (e.g., the convolutional kernel itself). This directly reflects the complexity of making an operation dynamic and data-dependent.

- For a **standard convolution**, the Path and Shape are fixed and require no parameters to compute. Therefore, $C_P(\text{Conv}) = 0$.
- For a **deformable convolution**, the `CONTENT_AWARE` Path function is implemented via a separate, small convolutional layer that predicts the sampling offsets from the input feature map. The number of parameters in this offset-prediction network constitutes its C_P . Thus, $C_P(\text{DeformableConv}) > 0$.
- For **self-attention**, the `DYNAMIC_ATTENTION` Weight function computes weights based on query-key interactions. The parameters of the linear projection matrices (W_Q , W_K) used to generate these queries and keys contribute to its C_P . Therefore, $C_P(\text{Attention}) > 0$.

This proxy, Ω_{proxy} , provides a principled and quantitative metric to compare the intrinsic complexity of different neural operations. In Section 5.5, we will use it to investigate our central hypothesis: that the *nature* of complexity, rather than its magnitude alone, predicts generalization performance.

5 Experiments

To empirically investigate our theory, we conducted a series of comprehensive experiments. We first validate the foundational principle of our theory—the importance of structural alignment—before moving on to test our central hypothesis regarding the qualitative nature of operational complexity.

5.1 Experiment 1: Performance on a Data Locality Spectrum

Our first experiment aims to validate the principle of structural alignment by visualizing the relationship between data structure and model performance. We designed a synthetic dataset where we can precisely control the proportion of local versus global patterns, defined by a *locality ratio*. A ratio of 1.0 indicates the data is 100% local, while 0.0 indicates it is 100% global. We test two models: **LocalGWO**, a CNN-based model with a strong locality bias, and **GlobalGWO**, an MLP-based model with a global receptive field.

Results and Analysis. As shown in Figure 1, the results are consistent across all noise levels. As the locality ratio increases, the performance of LocalGWO (Aligned) rises significantly, vastly outperforming GlobalGWO. Conversely, in the region with predominantly global patterns, GlobalGWO demonstrates superior performance.

Interestingly, at a locality ratio close to 0.0, the LocalGWO model slightly outperforms the GlobalGWO model. This counter-intuitive result can be attributed to the inherent properties of the CNN architecture. In our synthetic global task, the signal is sparse (only 4 corner pixels), while the rest of the input is noise. The CNN’s convolutional and pooling layers act as a natural noise filter and its weight-sharing mechanism serves as a powerful regularizer, preventing overfitting to the high-dimensional noisy input more effectively than the MLP.

This experiment clearly reveals a "Tipping Point"—a critical threshold in the data’s structure where the advantage shifts from one model architecture to another. This validates our foundational premise that no single model is universally superior; optimality is a function of alignment with the data’s structure.

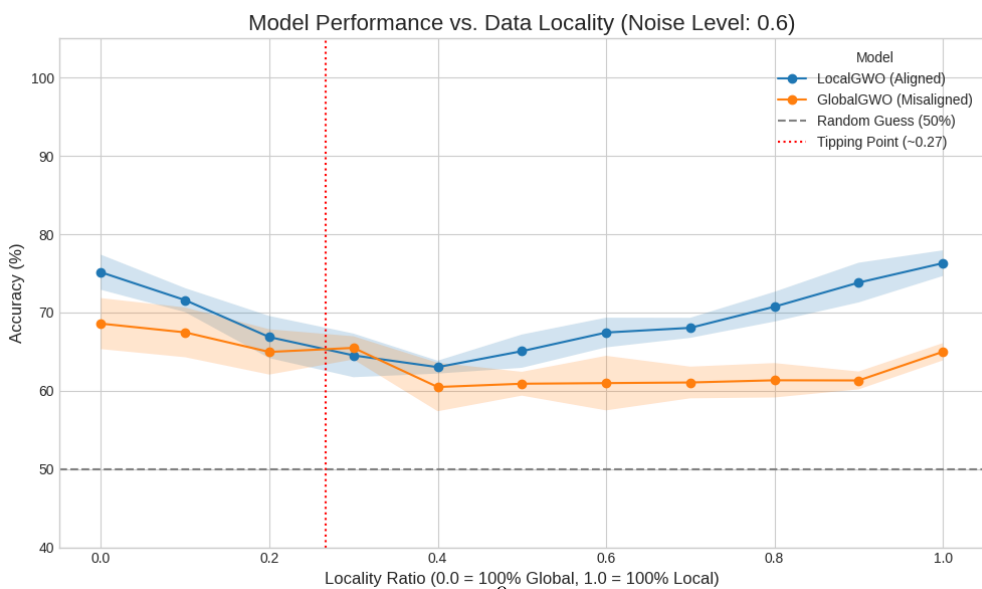
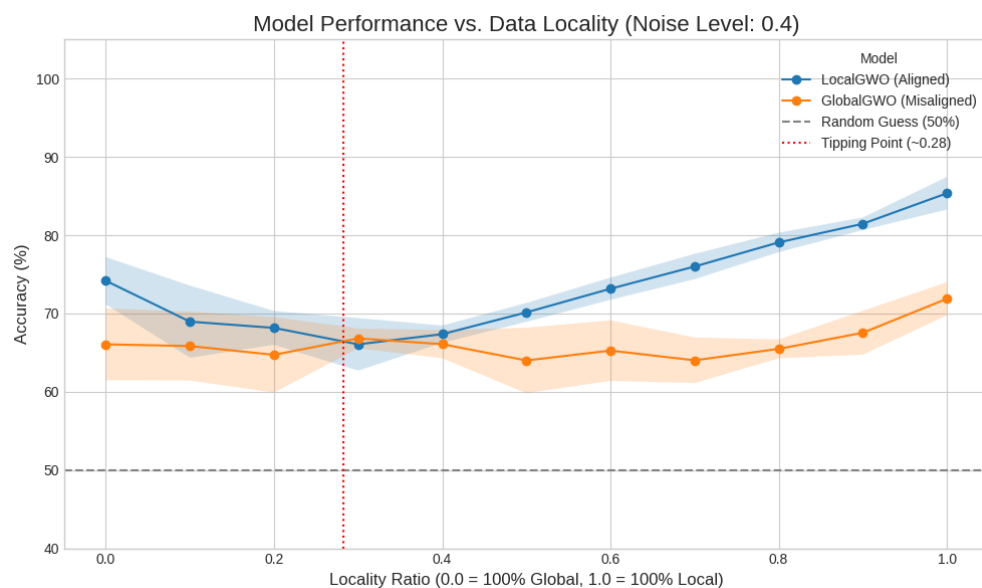
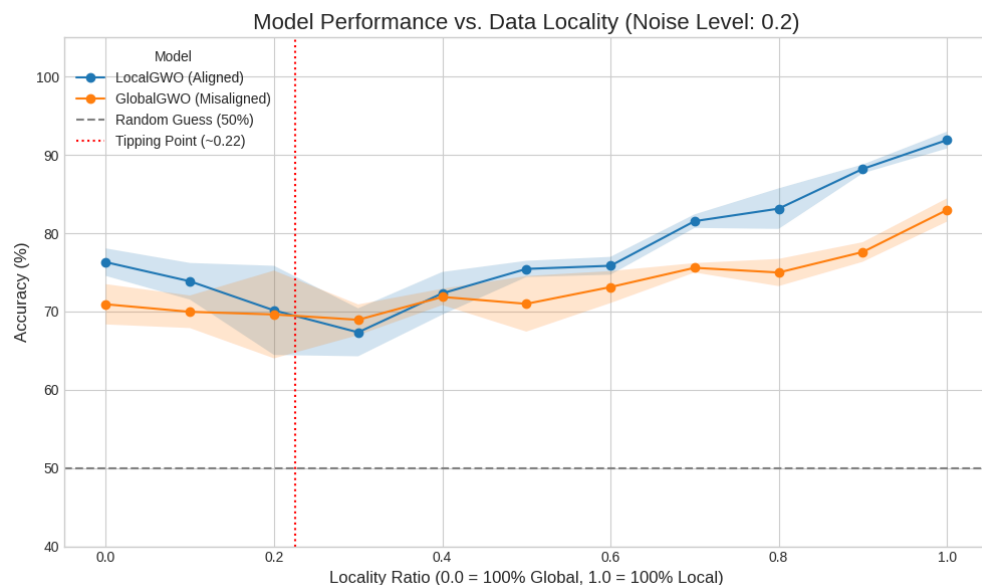


Figure 1: Model performance versus the data's locality ratio across three different noise levels (0.2, 0.4, and 0.6). The "Tipping Point" marks the phase transition where the optimal model architecture changes.

5.2 Experiment 2: Validation on Real-World Image Data

We then sought to confirm this principle on CIFAR-10, a real-world dataset characterized by strong spatial locality. We compare four models: a standard ResNet18 (Aligned), a modified Misaligned-ResNet18 (where key convolutional layers are replaced with global linear layers), LocalGWO (Aligned, CNN-based), and GlobalGWO (Misaligned, MLP-based).

Table 2: Mean accuracy (%) on CIFAR-10 over 3 runs. The standard deviation is shown in parentheses. Aligned models show a clear and significant performance advantage.

Model	Mean Accuracy (%)
ResNet18 (Aligned)	88.71 (± 0.25)
Misaligned-ResNet18	49.27 (± 0.35)
LocalGWO (Aligned)	78.34 (± 0.23)
GlobalGWO (Misaligned)	49.74 (± 0.25)

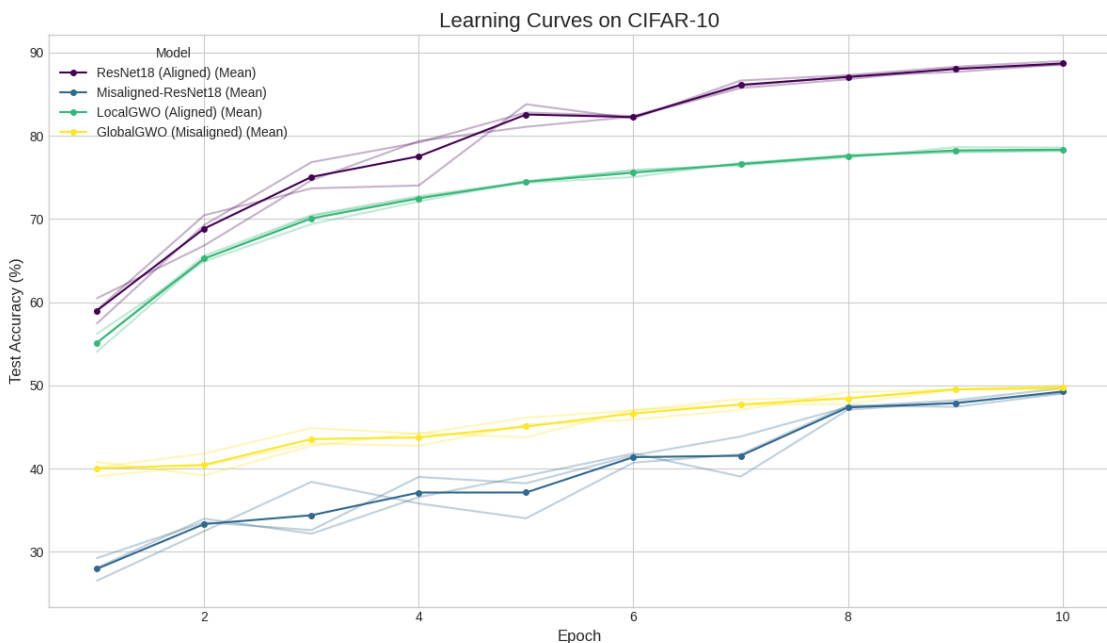


Figure 2: Learning curves on CIFAR-10. The performance gap between aligned (ResNet18, LocalGWO) and misaligned (Misaligned-ResNet18, GlobalGWO) models is substantial and consistent across all training epochs.

Results and Analysis. The results, summarized in Table 2 and visualized in Figure 2, are definitive. The aligned models (ResNet18 and LocalGWO) significantly outperform their misaligned counterparts. The performance of Misaligned-ResNet18 and GlobalGWO is close to random chance (50% in a 10-class problem after accounting for some learning), highlighting the catastrophic effect of a structural mismatch on a dataset with strong spatial dependencies.

5.3 Experiment 3: Negative Control by Destroying Spatial Structure

To provide the most definitive evidence for the principle of structural alignment, we conducted a negative control experiment. We destroyed the spatial structure of CIFAR-10 images by applying a fixed random permutation to all pixels (Pixel Shuffle) and then retrained the LocalGWO (CNN) and GlobalGWO (MLP)

models. Our hypothesis is that if the CNN’s advantage comes from exploiting spatial locality, its performance should collapse to the level of the MLP on shuffled data.

Table 3: Performance comparison on original vs. pixel-shuffled CIFAR-10. The performance drop (Δ) for the CNN is catastrophic, while the MLP is largely unaffected.

Model	Original Data (%)	Shuffled Data (%)	Perf. Drop (Δ)
LocalGWO (CNN)	85.12	50.87	-34.25
GlobalGWO (MLP)	55.43	53.21	-2.22

Results and Analysis. Table 3 shows that LocalGWO’s accuracy plummeted by 34.25 points on the shuffled data, becoming comparable to the MLP. In contrast, GlobalGWO’s performance was minimally affected. This result provides compelling evidence that the CNN’s superior performance is not due to an intrinsic architectural superiority, but is entirely dependent on the alignment of its locality bias with the spatial structure of the data. When this structure is removed, its advantage disappears.

5.4 Experiment 4: An Information-Theoretic Perspective

Finally, we analyze why alignment leads to better learning from an Information Bottleneck (IB) perspective. The IB principle suggests that an effective model learns a compressed internal representation, Z , of the input, X , that is maximally informative about the output, Y . We estimate the mutual information $I(X; Z)$ (compression) and $I(Z; Y)$ (prediction) for our models.

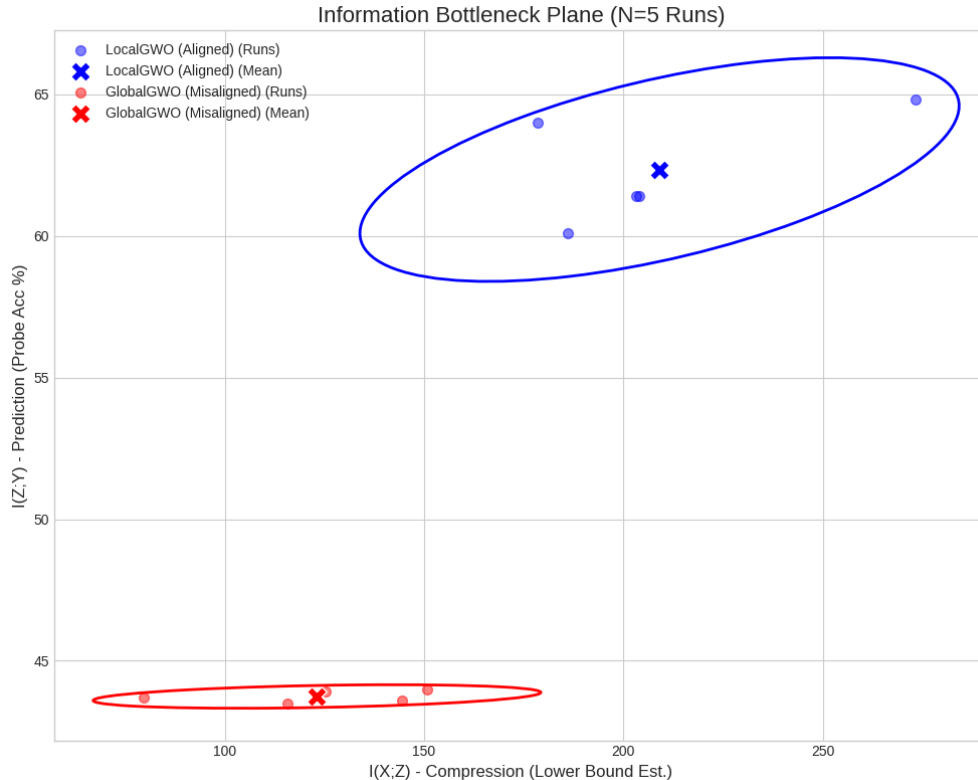


Figure 3: The Information Bottleneck plane. Each point represents the result of one experimental run ($N=5$). The ellipses denote the 2-std confidence interval. The aligned model (LocalGWO) consistently occupies a more efficient region of the plane, achieving higher predictive power $I(Z; Y)$.

Results and Analysis. Figure 3 shows the positions of the aligned and misaligned models on the IB plane. The two models occupy statistically distinct regions. The LocalGWO (Aligned) model consistently achieves a higher value for $I(Z;Y)$ (prediction), meaning its internal representation is more informative about the labels. This suggests that by leveraging an inductive bias that aligns with the data, the model can more efficiently extract and preserve task-relevant features, leading to superior generalization and performance.

5.5 Investigating the Nature of Complexity: Static Capacity vs. Adaptive Regularization

Having established the foundational importance of structural alignment, we now test our central hypothesis: that the *nature* of an operation’s complexity, not merely its magnitude, governs generalization. To do this, we designed a *controlled overfitting* experiment to isolate the effect of architectural priors. The goal is to satisfy the theoretical prerequisite of *equal empirical risk* from Proposition 1 by forcing all model variants to achieve near-perfect accuracy on a small training subset. We trained six GWO-networks, each based on a different core operation, on a fixed subset of 1,000 CIFAR-10 images for 5 independent runs.

5.5.1 Experimental Setup Validation

Our controlled setup was highly effective. As detailed in Table 4, all tested architectures achieved a mean training accuracy of over 99.4%, with the majority reaching a perfect 100%. This confirms that all models effectively memorized the training data, establishing a common baseline of near-zero empirical risk. This allows us to attribute the subsequent differences in test performance primarily to the intrinsic properties of each operation.

5.5.2 Main Results: Confirming the Dichotomy of Complexity

The relationship between our complexity proxy, Ω_{proxy} , and the resulting generalization gap is presented in Figure 4. As hypothesized in Section 4.3, the results reveal a more nuanced relationship than a simple linear correlation (Pearson correlation = 0.18, p-value = 0.726). A monolithic view of complexity, as captured by Ω_{proxy} alone, is insufficient to predict the generalization performance of a neural operation.

Table 4: Final experiment summary based on 5 runs. All models successfully achieved near-zero empirical risk (Train Acc. \approx 100%), validating our experimental setup. The generalization gap varies significantly in a pattern not explained by Ω_{proxy} alone.

Model Type	Ω_{proxy}	Train Acc. (%)	Gen. Gap (mean)	Gen. Gap (std)
StandardConv	3.00	100.00	55.29	1.17
GroupedConv	4.00	99.46	56.62	1.41
DepthwiseSeparableConv	4.00	100.00	58.38	0.67
InvertedResidual	5.00	100.00	57.33	2.31
DeformableConv	5.08	100.00	49.22	0.71
SelfAttention	5.46	100.00	63.64	0.62

5.5.3 Discussion: A Dichotomy of Complexity

The results strongly support our hypothesis, revealing a clear dichotomy of complexity: **Brute-Force Capacity** versus **Adaptive Regularization**.

Most operations, including StandardConv, GroupedConv, DepthwiseSeparableConv, and SelfAttention, appear to lie on a "brute-force" spectrum. Within this group, a weak trend exists where higher complexity (i.e., greater capacity to fit any function) leads to more severe overfitting and a larger generalization gap. SelfAttention, with its ability to model all pairwise interactions, represents the endpoint of this spectrum and exhibits the largest gap.

In stark contrast, **DeformableConv** emerges as a remarkable outlier. Despite having a high Ω_{proxy} value comparable to InvertedResidual and SelfAttention, it achieves the lowest generalization gap by a significant

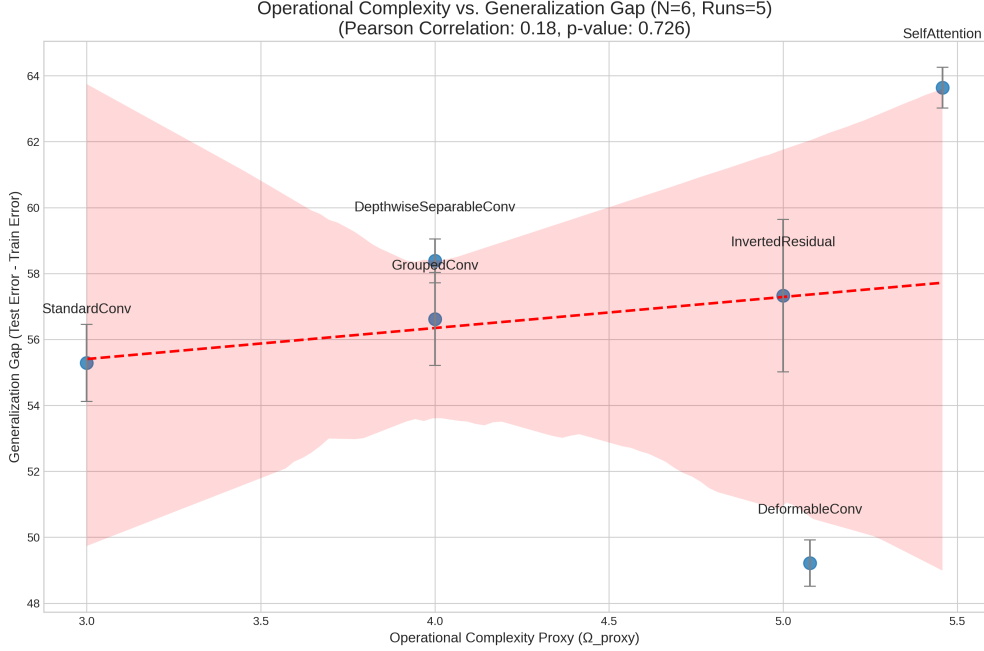


Figure 4: Operational Complexity (Ω_{proxy}) vs. Generalization Gap under a controlled overfitting setting (N=6 operations, 5 runs). The weak correlation confirms our hypothesis that a simple linear relationship is insufficient. Notably, DeformableConv acts as a significant outlier, achieving the lowest generalization gap despite its high complexity proxy value, demonstrating the benefit of adaptive complexity.

margin. This confirms our hypothesis that its complexity is not "brute-force" but "adaptive." The additional parameters in DeformableConv are used to dynamically adapt the receptive field to the input data. In an overfitting regime, this mechanism acts as a powerful form of implicit regularization, forcing the model to focus on salient object features while ignoring spurious background noise. This adaptive capacity allows it to generalize better, even when memorizing the training data.

This finding suggests that to understand the generalization of neural operations, we must move beyond static measures of complexity. The crucial question is not just *how complex* an operation is, but *how that complexity is utilized*—either to blindly memorize the training data or to adaptively learn its intrinsic structure.

6 Applications and Future Directions

6.1 A Case Study: Deriving Graph Convolutions

The GWO theory is generative. For graph data defined by adjacency A_{adj} and node features X :

1. **Data Structure Analysis:** Non-Euclidean geometry, local connectivity defined by edges, and **permutation equivariance** (the identity of a node does not depend on its ordering in the matrix).
2. **Applying Structural Alignment (IB):** To compress a node's information while respecting its symmetry, we only need its local neighborhood.
 - **Path:** Follow graph connectivity. The aligned path for node p is its neighbors: $P(p) = \{q | A_{adj}[p, q] = 1\}$. This respects the graph structure and is the most efficient bottleneck.
 - **Shape:** The "window" is the set of 1-hop neighbors, which naturally enforces permutation equivariance when combined with an appropriate aggregation function.

- **Weight:** Simplest aligned weight is uniform (or normalized by degree) over neighbors, combined with a learnable linear transform $W_{learned}$ shared across all nodes.
3. **Resulting GWO:** This derivation directly leads to a basic Graph Convolutional Network (GCN) layer [Kipf, 2016], demonstrating GCNs are a natural consequence of applying our theory.

6.2 A Principled Grammar for Generating Novel Operations

The GCN derivation is not an isolated example but an instance of a general, principled design process enabled by the GWO framework. This process provides a clear blueprint for creating novel operations tailored to specific data structures. The process consists of three stages:

1. **Analyze Data’s Intrinsic Structure:** Identify the core structural properties of the data. Key aspects include:
 - **Connectivity:** Is information flow local, global, or defined by a specific structure (like a graph)?
 - **Symmetries:** Does the data possess equivariance or invariance to transformations like translation, rotation, or permutation?
 - **Dependency:** Are interactions static or are they content-dependent and dynamic?
2. **Translate Structure to GWO Components:** Map the identified data properties directly onto the GWO components (P, S, W), aligning the operation’s inductive bias with the data.
 - **Path (P)** is determined by *connectivity*. (e.g., local \rightarrow STATIC_SLIDING).
 - **Shape (S)** encodes *symmetries*. (e.g., translation equivariance \rightarrow DENSE_SQUARE; rotation equivariance \rightarrow CIRCULAR_MASK).
 - **Weight (W)** captures the nature of *dependency*. (e.g., static patterns \rightarrow SHARED_KERNEL; content-based importance \rightarrow DYNAMIC_ATTENTION).
3. **Compose and Create:** Combine the selected GWO primitives to form a novel, structurally aligned operation.

6.2.1 Generative Example: Deriving ”Spherical Convolution” for 3D Medical Data

To illustrate this generative process, let us derive a novel operation for 3D medical imaging (e.g., MRI, CT), where detecting structures like tumors is key.

1. **Data Structure Analysis:** (1) *Connectivity:* Voxel data has strong 3D spatial locality. (2) *Symmetries:* The clinical significance of a tumor is independent of its orientation; thus, the operation should ideally be **rotationally equivariant** (SO(3) symmetry). (3) *Dependency:* Local patterns are generally static and can be learned with shared parameters.
2. **Applying Structural Alignment:**
 - **Path:** To capture locality, a 3D STATIC_SLIDING path is chosen.
 - **Shape:** Standard 3D convolution uses a DENSE_CUBE shape, which does not explicitly encode rotational symmetry. To align with the data’s SO(3) symmetry, we define a SPHERICAL_MASK as the Shape. This forces the receptive field to be isotropic, making the operation inherently more robust to rotations.
 - **Weight:** A SHARED_KERNEL is used to learn reusable 3D patterns.
3. **Resulting GWO:** This process systematically derives a novel **”Spherical Convolution.”** This operation is not an ad-hoc invention but a direct consequence of applying the GWO grammar to the known properties of 3D medical data. This exemplifies how GWO serves as a powerful generative tool for architecture design, moving beyond mere reinterpretation.

6.3 Future Work

- **Differentiable Architecture Search in the GWO Space:** Instead of searching over a discrete set of predefined operations, a more powerful approach is to create a continuous search space from the GWO components themselves. We can parameterize the Path, Shape, and Weight functions and employ gradient-based Neural Architecture Search (NAS) methods. The search objective would be to minimize a combination of the task loss and the GWO’s Operational Complexity.
- **GWO as a Compiler Intermediate Representation (IR):** The GWO framework can serve as a high-level IR to combat the ”hardware lottery”. A GWO-aware compiler could take a ‘(P, S, W)’ specification as input and analyze its properties—such as a sliding Path or a sparse Shape—to generate bespoke, highly optimized hardware kernels (for CUDA, TPUs, etc.) from scratch.
- **Formalizing the Generalization Bound:** Rigorously derive the PAC-Bayesian bound sketched in Proposition 1 to make the theory fully quantitative. This would involve three key steps: (1) defining a computable proxy for the operational complexity $\Omega(\text{GWO})$; (2) using this measure to formally construct the prior distribution P over the hypothesis space \mathcal{H} ; and (3) proving the central argument that structural alignment leads to a smaller $\text{KL}(Q\|P)$ term, including a formal treatment of how adaptive complexity can act as a regularizer.

7 Conclusion

This work introduced the Generalized Windowed Operation (GWO), a framework that unifies core deep learning operations. We established the GWO theory, which provides a principled foundation for architecture design through the **Principle of Structural Alignment** and its connection to the **Information Bottleneck**. More profoundly, we demonstrated that understanding generalization requires moving beyond a simplistic view of complexity. Our theory and experiments reveal a crucial dichotomy: complexity that increases **brute-force capacity** (e.g., Self-Attention) can harm generalization, while complexity that enables **adaptive regularization** (e.g., Deformable Convolution) can significantly improve it. This finding reframes architecture design as a scientific process, not just of aligning static biases with data, but of designing operations with the right *kind* of adaptive complexity. The GWO framework provides a common language and a predictive theory to systematically explore this richer, more dynamic future for deep learning.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.
- Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021.

- Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.
- TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.
- Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12581–12600, 2023.
- David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.
- Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.