

000

001

002

003

004

005

006

007

Anonymous authors

Paper under double-blind review

009

010

011

012

ABSTRACT

013

014

015

016

017

018

019

020

021

022

023

024

025

026

027

028

029

1 INTRODUCTION

030

Transformers (Vaswani et al., 2017) have become widespread in deep learning, demonstrating unprecedented success on a massive scale (Dosovitskiy et al., 2021; Jumper et al., 2021; Devlin et al., 2019). Their power lies in simple, general-purpose mechanisms that have matured over the years and continue to offer remarkable gains in speed and flexibility, benefiting from vast datasets and computational resources. Yet, this very generality implies they are not inherently equipped to handle specific symmetries present in many scientific domains. For problems with geometric structure, such as those in physics, molecular chemistry, and 3D computer vision, performance can be significantly enhanced by incorporating such inductive bias (Fuchs et al., 2020; Ying et al., 2021; Zhao et al., 2021; Bekkers et al., 2024; Balla et al., 2024; Liao et al., 2024; Romero & Cordonnier, 2021; Wessels et al., 2024; Bose et al., 2024; Zhdanov et al., 2024; Nyholm et al., 2025). The principle of symmetry, for example, has given rise to highly data-efficient and robust group equivariant networks (Cohen & Welling, 2016; 2017; Cesa et al., 2022). However, scaling these symmetry-aware networks has been difficult, as their reliance on operations like group convolutions or Clebsch-Gordan tensor products introduces significant computational overhead compared to standard architectures (He et al., 2021a; Luo et al., 2024). This raises the question: *how can we leverage powerful geometric inductive biases within the transformer architecture without sacrificing the speed and flexibility integral to its success?*

046

047

048

049

050

051

052

053

A central challenge in addressing this problem lies in designing an attention mechanism that inherently respects geometric transformations. Such a mechanism would expand on the inductive bias of Transformers, which is typically limited to position embeddings. While widely-used, absolute positional encodings provide location information, but they enforce no explicit relational structure (Shaw et al., 2018; He et al., 2021b). A significant step towards this goal has been the adoption of Rotary Position Embeddings (RoPE) (Su et al., 2024), which endows attention with translation equivariance. Yet, extending this to roto-translation equivariance within the standard Transformer framework remains challenging. Existing approaches often achieve this by making complex architectural changes to equivariant networks that poorly scale or settle for invariant attention mechanisms

which sacrifice feature representations for simplicity and computational efficiency (Masters et al., 2022; Assaad et al., 2023; Thölke & Fabritiis, 2022; Brehmer et al., 2023; Kundu & Kondor, 2025; Joshi et al., 2025). Recent efforts have also explored *hybrid architectures* that resort to symmetry breaking (Qu & Krishnapriyan, 2024; Lawrence et al., 2025) to improve scalability but require a careful mix of modules to maximize downstream performance.

Our main contribution is the *Platonic Transformer*, a framework that achieves equivariance to continuous translations and discrete roto-reflections in Transformers *without changing the underlying attention mechanism or computation graph*. To achieve this, our method processes features relative to a collection of reference frames that form a Platonic symmetry group ($\mathcal{G} \subset O(3)$) and constrains all linear layers to be equivariant with respect to this choice of frame. This principled scheme allows the standard attention block, including its unmodified Rotary Position Embeddings (RoPE), to operate in parallel across these frames, and effectively associates each reference frame with a distinct attention head. As a result, the model incorporates a geometric inductive bias without altering the architecture or computational footprint of a standard Transformer. This enables flexible usage across domains at no additional cost, resolving the long-standing symmetry-awareness vs. scaling dilemma.

Additionally, we analyze the formal connection between RoPE-based attention and convolution to highlight its underlying inductive bias. We show that when the softmax operation is omitted, the attention becomes mathematically equivalent to a dynamic, content-aware convolution. Moreover, in this convolutional setting, the attention operator’s complexity scales linearly with the number of tokens, akin to methods like Performer (Choromanski et al., 2020). This result reframes RoPE-attention as a mechanism that explicitly learns and applies dynamic, content-aware geometric filters.

2 BACKGROUND: TRANSFORMERS WITH POSITION EMBEDDINGS

The core of a Transformer is its self-attention mechanism, which computes outputs for a sequence of input features $\{f_i \in \mathbb{R}^C\}$ based on pairwise interactions. To perform spatial tasks, this operation must incorporate the position $p_i \in \mathbb{R}^n$ associated with each feature f_i . This positional information, often added via absolute or relative encodings, allows the model to learn relationships that respect geometric symmetries.

2.1 VANILLA ATTENTION AND ABSOLUTE POSITIONING

Given a sequence of input features $f_i \in \mathbb{R}^C$, the self-attention layer first computes query, key, and value vectors via linear projections: $\mathbf{q}_i = \mathbf{W}^Q f_i$, $\mathbf{k}_j = \mathbf{W}^K f_j$, $\mathbf{v}_j = \mathbf{W}^V f_j$. Here, the learnable weight matrices are $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{C \times d}$ and $\mathbf{W}^V \in \mathbb{R}^{C \times C'}$. The output for the i -th feature, $\mathbf{y}_i \in \mathbb{R}^{C'}$, is a weighted sum of the value vectors, with weights determined by softmax-normalized dot products of queries and keys:

$$\mathbf{y}_i = \sum_{j=1}^N \text{attn}(\mathbf{q}_i, \mathbf{k}_j) \mathbf{v}_j, \quad \text{where} \quad \text{attn}(\mathbf{q}_i, \mathbf{k}_j) = \text{softmax}_j \left(\mathbf{q}_i^\top \mathbf{k}_j \right). \quad (1)$$

As this operation is permutation-equivariant, it is insensitive to the order of the inputs and must be modified to incorporate positional information for spatial tasks. A common approach is to use Absolute Positional Encodings (APE), where a unique vector $\mathbf{E}(\mathbf{p}_i)$ is added to each input feature, $\mathbf{f}'_i = \mathbf{f}_i + \mathbf{E}(\mathbf{p}_i)$, before the linear projections are applied. The attention score is then computed from these position-aware features. However, since this interaction depends on absolute coordinates rather than relative positions, APE is not translation-equivariant.

2.2 ROTARY POSITION EMBEDDINGS (ROPE)

RoPE achieves a more structured approach to position encoding (Su et al., 2024). Instead of adding a positional vector, RoPE modifies the query and key vectors with a position-dependent transformation, making the attention score explicitly dependent on relative positions.

This transformation is constructed by stacking 2D rotation matrices, giving RoPE its name. To apply RoPE with positions \mathbf{p} in dimension $n > 1$, we use a set of n -dimensional frequency vectors

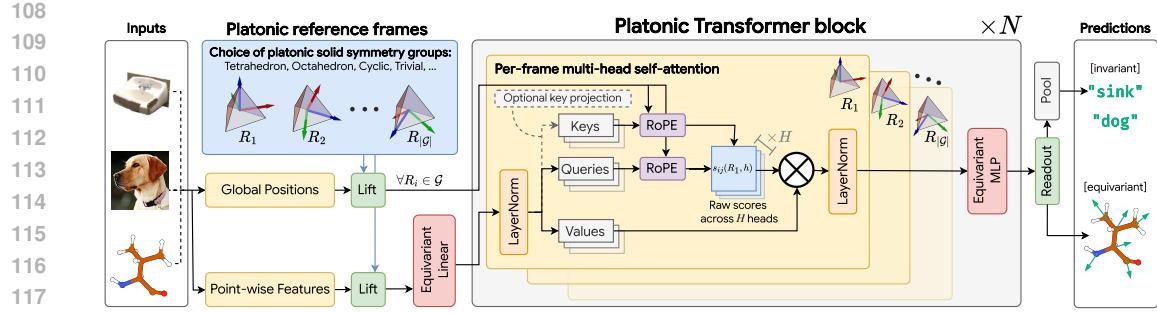


Figure 1: Visualization of Weight-Shared RoPE within the N -layer Platonic Transformer. Scalar and vector inputs are lifted to become functions on the platonic solid symmetry group of choice (here, the Tetrahedral group). The same multi-head self-attention mechanism is applied in parallel, with each instance rotating the features according to a different reference frame $R_i \in \mathcal{G}$. Choosing the trivial group as \mathcal{G} reduces this framework to a standard Transformer.

$\Omega = \{\omega_k\}_{k=1}^{d/2}$, each defining a direction used to project \mathbf{p} to 1D and a frequency used to apply 1D-RoPE in this direction. We obtain $d/2$ blocks,

$$\rho_{\omega_k}(\mathbf{p}) = \begin{pmatrix} \cos(\omega_k^\top \mathbf{p}) & -\sin(\omega_k^\top \mathbf{p}) \\ \sin(\omega_k^\top \mathbf{p}) & \cos(\omega_k^\top \mathbf{p}) \end{pmatrix}, \quad (2)$$

which are stacked in a block-diagonal manner to form a single transformation matrix, $\rho_\Omega(\mathbf{p})$:

$$\rho_\Omega(\mathbf{p}) = \text{diag}(\rho_{\omega_1}(\mathbf{p}), \dots, \rho_{\omega_{d/2}}(\mathbf{p})). \quad (3)$$

Note that while $\rho_\Omega(\mathbf{p})$ is a high-dimensional rotation, this rotation is not related to rotations of the position \mathbf{p} . In fact, ρ_Ω is instead connected to translations of \mathbf{p} , formally discussed in Appendix A.

For a query \mathbf{q}_i at position \mathbf{p}_i and a key \mathbf{k}_j at position \mathbf{p}_j , ρ_Ω is applied before the dot product. As the operator ρ_Ω is orthogonal and satisfies the homomorphism property¹ for translations, the interaction simplifies to depend only on relative positions:

$$(\rho_\Omega(\mathbf{p}_i)\mathbf{q}_i)^\top (\rho_\Omega(\mathbf{p}_j)\mathbf{k}_j) = \mathbf{q}_i^\top \rho_\Omega(\mathbf{p}_i)^\top \rho_\Omega(\mathbf{p}_j)\mathbf{k}_j = \mathbf{q}_i^\top \rho_\Omega(\mathbf{p}_j - \mathbf{p}_i)\mathbf{k}_j. \quad (4)$$

This final form reveals the core property of RoPE. Although widely adopted for its empirical success, the mechanism's effectiveness is not coincidental; it directly embeds translation equivariance into the attention mechanism by making the score a function of content and relative positions. This powerful geometric inductive bias, often hidden within the standard Transformer framework, provides a principled reason for RoPE's strong performance (Chen et al., 2023; Dai et al., 2019). The formal construction of this operator from the first principles of group theory is detailed in Appendix A.

3 THE PLATONIC TRANSFORMER

We generalize the principle of RoPE to obtain equivariance not only under continuous translations, but also discrete roto-reflections. We obtain roto-reflection equivariance by redefining the positional encoding relative to a set of reference frames defined as elements in a discrete subgroup $\mathcal{G} \subset O(n)$. Traditional RoPE-attention operates on a single global reference frame. Instead, we perform attention on multiple frames in parallel. A key advantage of our method is that it leaves the rope-attention mechanism and the overall computation graph unchanged from the traditional transformer.

3.1 FEATURES RELATIVE TO REFERENCE FRAMES

Throughout the architecture, features are represented and processed *relative to the reference frames* defined by the elements of a discrete group $\mathcal{G} \subset O(n)$. Since input features are typically defined in a

¹The operator $\rho_\Omega(\mathbf{p})$ being orthogonal means its inverse is its transpose: $\rho_\Omega(\mathbf{p})^{-1} = \rho_\Omega(\mathbf{p})^\top$. The homomorphism property for the translation group satisfies $\rho_\Omega(\mathbf{p}_i + \mathbf{p}_j) = \rho_\Omega(\mathbf{p}_i)\rho_\Omega(\mathbf{p}_j)$.

162 global frame of reference, they must first be *lifted* to become functions on the group \mathcal{G} . Specifically,
 163 each feature becomes a map $\mathbf{f}_i(\cdot) : \mathcal{G} \rightarrow \mathbb{R}^C$, where $\mathbf{f}_i(R)$ is the feature vector at point i viewed from
 164 frame $R \in \mathcal{G}$. For the finite groups we consider, this map is represented as a tensor of shape $[\mathcal{G}, C]$.
 165 We denote this tensor simply as a flattened vector $\mathbf{f}_i \in \mathbb{R}^{|\mathcal{G}| \cdot C}$ and use the functional notation $\mathbf{f}_i(\cdot)$ to
 166 emphasize its role as a feature map. As we will see, the flattened vector viewpoint is key to preserving
 167 the standard Transformer computation graph.

168 The lifting process depends on the geometric type of the input feature. Scalar features, being invariant
 169 to viewpoint, are lifted to constant functions by copying them across all frames. Vector features, in
 170 contrast, are expressed relative to each frame; for example, a single 3D vector feature $\mathbf{u} \in \mathbb{R}^3$ is
 171 lifted to a three-channel signal on the group via the transformation $\mathbf{f}(R) = R^{-1}\mathbf{u}$. All such lifted
 172 components can be concatenated, after which they are processed by the subsequent equivariant,
 173 frame-dependent attention layers.

175 3.2 WEIGHT-SHARING ACROSS RoPE EMBEDDINGS

177 The key step for achieving equivariance to \mathcal{G} as well as translations is making the RoPE operator itself
 178 dependent on the reference frames. This is achieved by projecting the position \mathbf{p}_i of each input token
 179 i onto R , which yields views $\mathbf{p}_i(R) = R^{-1}\mathbf{p}_i$ relative to each frame. As the queries \mathbf{q}_i , keys \mathbf{k}_j , and
 180 values \mathbf{v}_j are obtained by applying equivariant linear projections (cf. Section 3.3) to the feature maps
 181 \mathbf{f}_i , they are also functions on the group. We can then compute the unnormalized attention scores from
 182 the perspective of frame R , which we denote as $s_{ij}(R)$:

$$s_{ij}(R) = \mathbf{q}_i(R)^\top \rho_\Omega(\mathbf{p}_j(R) - \mathbf{p}_i(R)) \mathbf{k}_j(R) \quad (5)$$

$$= \mathbf{q}_i(R)^\top \rho_\Omega((\mathbf{p}_j - \mathbf{p}_i)(R)) \mathbf{k}_j(R). \quad (6)$$

185 Scores for each frame are computed *in parallel* as their own independent attention head. Note that we
 186 can also obtain $s_{ij}(R)$ by steering the base set of frequencies Ω instead of the positions \mathbf{p}_i , which we
 187 show in Appendix D. However, from our current perspective, the RoPE-attention mechanism itself
 188 remains completely unchanged from its traditional formulation in Eq. 4; only the relative positions
 189 $\mathbf{p}_i - \mathbf{p}_j$ are now defined relative to each reference frame R . The attention coefficients are obtained
 190 by applying the softmax to the scores $s_{ij}(R)$. The output $\mathbf{y}_i(R)$ for each token i is then given as

$$191 \quad \mathbf{y}_i(R) = \sum_{j=1}^N \text{attn}_{ij}(R) \mathbf{v}_j(R), \quad \text{where} \quad \text{attn}_{ij}(R) = \text{softmax}_j(s_{ij}(R)). \quad (7)$$

194 This process naturally results in an output tensor $\mathbf{y}_i \in \mathbb{R}^{|\mathcal{G}| \cdot C}$, where the features are defined relative
 195 to each frame. Notably, the base frequencies Ω of RoPE are shared across frames and this leads to
 196 the operator being equivariant to the roto-reflections in G , as we detailed in Appendix B.

198 3.3 EQUIVARIANT LINEAR LAYERS AND FIXED COMPUTATION GRAPH

200 All linear transformations, including the query, key, and value projections $(\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V)$, and
 201 any MLP blocks, must be equivariant. As our features can be viewed either as functions on the group,
 202 $\mathbf{f}_i(\cdot)$, or as flattened vectors, $\mathbf{f}_i \in \mathbb{R}^{|\mathcal{G}| \cdot C}$, we can describe the action of an equivariant linear layer Φ
 203 from both perspectives. From the flattened vector viewpoint, the layer is a standard matrix-vector
 204 multiplication, $\mathbf{y}_i = \mathbf{W}\mathbf{f}_i$. However, for this transformation to be equivariant, the weight matrix \mathbf{W}
 205 cannot be arbitrary; it must have a specific, constrained structure.

206 The equivariance constraint is defined from the functional viewpoint: for any group element $R \in \mathcal{G}$,
 207 the transformation must satisfy $\Phi(L_R \mathbf{f}_i) = L_R(\Phi(\mathbf{f}_i))$, where L_R is the action of rotating the
 208 reference frames, i.e., $(L_R \mathbf{f}_i)(\tilde{R}) = \mathbf{f}_i(R^{-1}\tilde{R})$. This constraint is satisfied *if and only if* the layer's
 209 operation is a *group convolution* (Cohen et al., 2019, Thm. 3.1). This gives the layer a dual
 210 identity: it is a convolution over the group axis, which is mathematically equivalent to a matrix-vector
 211 multiplication with a structured, weight-shared matrix:

$$212 \quad (\Phi(\mathbf{f}_i))(R) := \sum_{\tilde{R} \in \mathcal{G}} \mathbf{W}_{\text{group}}(R^{-1}\tilde{R}) \mathbf{f}_i(\tilde{R}) \quad \iff \quad \Phi(\mathbf{f}_i) := \mathbf{W}\mathbf{f}_i. \quad (8)$$

214 Here, $\mathbf{W}_{\text{group}} : \mathcal{G} \rightarrow \mathbb{R}^{C' \times C}$ is a learnable kernel defined on the group. The large matrix $\mathbf{W} \in$
 215 $\mathbb{R}^{(|\mathcal{G}| \cdot C') \times (|\mathcal{G}| \cdot C)}$ is a block matrix whose blocks are determined by the kernel values: $[\mathbf{W}]_{R, \tilde{R}} =$

216 $\mathbf{W}_{\text{group}}(R^{-1}\tilde{R})$. This structure imposes a weight-sharing scheme where the interaction between input
 217 and output frames depends only on their *relative pose*, $R^{-1}\tilde{R}$. The layer is thus constrained to learn
 218 patterns from the geometric arrangement of features, rather than their absolute pose.
 219

220 While the group convolution formulation makes the geometric inductive bias explicit, the matrix-
 221 vector viewpoint clarifies that this is in essence a principled *weight-sharing scheme* that preserves
 222 the computation graph of a standard linear layer (we’re still doing matrix-vector multiplication). A
 223 favorable side-effect, however, is that this structure reduces the parameter count from the $(|\mathcal{G}| \cdot C') \times$
 224 $(|\mathcal{G}| \cdot C)$ of an unconstrained layer to just $|\mathcal{G}| \cdot C' \cdot C$ —a reduction by a factor of $|\mathcal{G}|$.
 225

226 Crucially, by choosing the number of channels C such that the effective feature dimension $C \cdot |\mathcal{G}|$ is
 227 held constant, the overall matrix dimensions are identical regardless of the group size. The trivial
 228 group $\mathcal{G} = \{e\}$ illustrates the base case, where the operation collapses to a standard linear layer with
 229 a weight matrix $\mathbf{W}_{\text{group}}(e)$ of size $C' \times C$. The geometric inductive bias is therefore not introduced
 by adding new, complex modules, but by imposing a structure on the weights of existing ones.²
 230

231 With all components of the architecture now defined as equivariant operations, we can formally state
 232 the key property of the full model, namely equivariance under the discrete group $\mathcal{G} \subset O(n)$.
 233

234 **Proposition 1** (End-to-End Equivariance). *Our proposed Transformer architecture is an equivariant
 235 model. A global roto-reflection $R \in \mathcal{G}$ applied to the input point cloud results in a corresponding
 236 transformation L_R of the final output feature maps.*

237 The proof is given in Appendix B.
 238

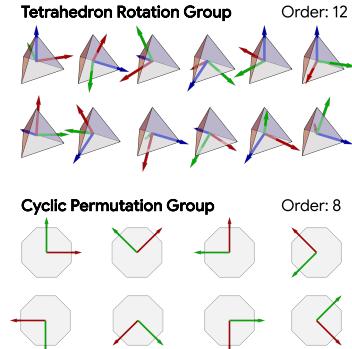
3.4 FRAME SELECTION VIA PLATONIC SOLIDS

239 The final step is to select a suitable subgroup $\mathcal{G} \subset O(n)$ to serve as the reference frames. We
 240 select them from the discrete symmetry groups of regular polygons and polyhedra, with different
 241 considerations for 2D and 3D as illustrated in Figure 2.
 242

243 In 3D, we restrict our frames to the finite *rotational* symmetry groups ($\mathcal{G} \subset SO(3)$) of the Platonic
 244 solids: the *tetrahedral* (12 rotations), *octahedral* (24 rotations), and *icosahedral* (60 rotations) groups.
 245 While these solids have larger full symmetry groups that include reflections (e.g., 24 total symmetries
 246 for the tetrahedron), we focus on the purely rotational subgroups for a more tractable structure.
 247

248 In 2D, we consider discrete subgroups of $O(2)$, which correspond to the symmetries of regular polygons. This includes both
 249 the rotation-only *cyclic groups* (C_n) and the *dihedral groups* (D_n), which contain both rotations and reflections. Here n denotes the group’s order. This discrete subgroup approach is
 250 advantageous for two reasons. First, it provides a finite set of frames that forms a structured and approximately uniform dis-
 251 cretization of the underlying continuous spaces of orientations
 252 ($SO(3)$ in 3D and $O(2)$ in 2D). Second, and more critically,
 253 these frames form a group. This is essential for maintaining a
 254 meaningful geometric structure, as it ensures that layers can op-
 255 erate equivariantly, keeping features coherently defined relative
 256 to our chosen frames throughout the network.
 257

258 The advantage of working with a *finite* group \mathcal{G} is that its op-
 259 erations can be handled discretely and efficiently using *Cayley*
 260 *tables*. We assign a unique index $i \in \{0, \dots, |\mathcal{G}| - 1\}$ to each
 261 rotation $R_i \in \mathcal{G}$. The group product $R_i R_j = R_k$ can then be
 262 precomputed and stored in the Cayley table, a simple look-up table where $\text{Cayley}[i, j] = k$. This
 263 discrete formalism makes the group action on our feature maps, which are functions on the group
 264 $f : \mathcal{G} \rightarrow \mathbb{R}^C$, extremely efficient. A rotation of this feature map by an element R_i , defined by the
 265 action $(L_{R_i} f)(R_j) = f(R_i^{-1} R_j)$, simplifies to a permutation of the feature tensor’s entries. With
 266



267 Figure 2: Elements of the symme-
 268 try groups of platonic solids form
 269 a subgroup of $SO(3)$.
 270

²This structure can even give computational benefits, by implementing the linear layers in the Fourier domain of \mathcal{G} (Bökman et al., 2025). In Appendix N, we find that at marginally higher channel counts than used in this paper, a Fourier implementation leads to greatly improved training throughput, indicating that this is a promising direction for future research.

270 the Cayley table, the new feature at position j is simply copied from the old feature at position
 271 $k = \text{Cayley}[\text{inverse}[i], j]$.
 272

273 4 INDUCTIVE BIAS OF PLATONIC TRANSFORMERS

276 This section examines the Platonic Transformer’s structural inductive biases. We highlight its
 277 interpretation as a dynamic group convolution and its equivariant attention, contrasting these with
 278 approaches based on invariant attention.

279 4.1 PLATONIC TRANSFORMER AS DYNAMIC GROUP CONVOLUTION

281 The use of RoPE in a linear attention setting establishes a deep connection to convolution. Specifically,
 282 the mechanism implements an adaptive convolution where the kernel is synthesized on-the-fly.
 283 This dynamic kernel is expressed as an expansion in a sparse Fourier basis, defined by the RoPE
 284 frequencies, and the coefficients for this basis expansion are provided by the query vectors. This
 285 makes the convolution content-aware. We formalize this as follows (proof in Appendix C.1).

286 **Proposition 2** (Linear RoPE Attention as Dynamic Convolution). *Consider a standard linear attention
 287 layer using RoPE with constant key vectors ($\mathbf{k}_j = \mathbf{1}$). The layer’s output \mathbf{y}_i is mathematically
 288 equivalent to a dynamic convolution:*

$$290 \quad \mathbf{y}_i = \sum_{j=1}^N \phi_{\mathbf{q}_i}(\mathbf{p}_j - \mathbf{p}_i) \mathbf{v}_j, \quad (9)$$

293 where the dynamic kernel $\phi_{\mathbf{q}_i}$ is given by the inverse sparse Fourier transform:

$$295 \quad \phi_{\mathbf{q}_i}(\Delta \mathbf{p}) = \sum_{k=1}^{d/2} [a_k(\mathbf{q}_i) \cos(\omega_k^\top \Delta \mathbf{p}) + b_k(\mathbf{q}_i) \sin(\omega_k^\top \Delta \mathbf{p})]. \quad (10)$$

298 The Fourier coefficients are given by the linear projections $a_k(\mathbf{q}_i) = q_{i,2k-1} + q_{i,2k}$ and $b_k(\mathbf{q}_i) =$
 299 $q_{i,2k} - q_{i,2k-1}$, where $q_{i,m}$ is the m -th element of the query vector \mathbf{q}_i .

300 **Remark 1** (Purely Geometric vs. Mixed Kernels). *This result recasts the query’s role: rather than
 301 simply probing for content, \mathbf{q}_i enables the parameters to construct a unique geometric filter. The
 302 formulation of the key vector is a design choice. The constant-key formulation ($\mathbf{k}_j = \mathbf{1}$) forces the
 303 model to learn purely geometric, content-adaptive convolution operators. In contrast, a learned key
 304 ($\mathbf{k}_j = \mathbf{W}^K \mathbf{f}_j$) results in a mixed kernel whose coefficients depend on both query and key features,
 305 and thus entangles geometry and signal.*

307 Our choice of the purely geometric formulation was motivated by a key experimental observation:
 308 training with a mixed kernel from learned keys was highly unstable on the QM9 and OMol25
 309 datasets (cf. Appendix L). We hypothesize this instability arises because these tasks require learning
 310 universal *physical principles* that are purely functions of geometry, whereas computer vision tasks like
 311 ScanObjectNN involve learning *statistical correlations* between local appearance and global shape. A
 312 mixed kernel *entangles* these universal principles with instance-specific features, creating an unstable
 313 optimization problem with *conflicting gradients* as the model attempts to learn a general physical
 314 law while simultaneously fitting unique local chemical environments. In computer vision, this same
 315 entanglement is beneficial, as learning the statistical interplay between features and geometry is the
 316 primary objective. Fixing the keys thus acts as a crucial regularizer for physical systems by forcing
 317 the model to prioritize the disentangled geometric principles, and thus stabilizing the training process.
 318 The convolution perspective further leads to a key practical advantage.

319 **Corollary 1** (Linear-Time Complexity). *The dynamic convolution in Proposition 2 can be computed
 320 in $O(N)$ time, where N is the number of tokens or points in the point cloud. This offers a scalable
 321 alternative to standard attention, which has a quadratic complexity of $O(N^2)$.*

322 Within our Platonic Transformer, this entire mechanism is lifted to operate over the reference frames
 323 defined by a group \mathcal{G} . Consequently, the operator becomes an adaptive *group convolution* (proof in
 Appendix C.3), where the kernel is steered by the group elements/reference frames.

324 4.2 INVARIANT VS. EQUIVARIANT ATTENTION SCORE
325

326 Our approach implements an *equivariant attention* mechanism, where the attention pattern is
327 orientation-dependent. This contrasts with methods using an *invariant attention* score, which applies
328 the same pattern from all orientations (Fuchs et al., 2020; Chen & Villar, 2022; Assaad et al., 2023;
329 Frank et al., 2024; Knigge et al., 2024; Kundu & Kondor, 2025; Nordström et al., 2025).

330 For multi-head attention with H heads, let $\mathbf{q}_i(R, h)$, $\mathbf{k}_j(R, h)$, and $\mathbf{v}_j(R, h)$ denote the projected
331 query, key, and value vectors for head h from the perspective of frame R . In our equivariant approach,
332 the raw scores $s_{ij}(R, h)$ are passed directly to the softmax. This allows the model to learn orientation-
333 dependent attention patterns, making it a more expressive formulation that retains the rich geometric
334 information in the features. The output is an equivariant feature map on the group:

$$335 \quad \mathbf{y}_i(R, h) = \sum_{j=1}^N \text{softmax}(\underbrace{s_{ij}(R, h)}_{R-\text{dependent}}) \mathbf{v}_j(R, h), \quad s_{ij}(R, h) = \mathbf{q}_i(R, h)^\top \boldsymbol{\rho}_{\Omega_h}((\mathbf{p}_j - \mathbf{p}_i)(R)) \mathbf{k}_j(R, h). \quad (11)$$

339 In practice, this is efficiently implemented by treating the $|\mathcal{G}|$ perspectives as an independent set of
340 attention heads. Tensors are reshaped so that the group and head dimensions are merged, e.g., to a
341 shape of $[B, N, |\mathcal{G}| \cdot H, C_h]$, before the dot product calculation.

342 In an invariant attention score, a single attention pattern is created by pooling the raw scores over the
343 group axis before the softmax, akin to the *symmetrization* in the RoPE-based approach of Frank et al.
344 (2024). These invariant attention scores are then applied to the original equivariant value vectors. The
345 resulting output is still equivariant, but it is derived from an *orientation-agnostic attention pattern*:

$$347 \quad \mathbf{y}_i(R, h) = \sum_{j=1}^N \text{softmax}(\underbrace{s_{ij}^{\text{inv}}(h)}_{R-\text{agnostic}}) \mathbf{v}_j(R, h), \quad \text{where} \quad s_{ij}^{\text{inv}}(h) = \sum_{R \in \mathcal{G}} s_{ij}(R, h). \quad (12)$$

350 Although simpler, this formulation sacrifices the model’s ability to attend to features in an orientation-
351 dependent manner. Implementing Eq. 12 can be done by reshaping tensors so that the group and
352 channel dimensions are merged, to shape $[B, N, H, |\mathcal{G}| \cdot C_h]$, as then the dot-product in s_{ij} and the
353 sum in s_{ij}^{inv} are simultaneously computed when taking the dot-product between queries and keys. For
354 a fully invariant output, one could additionally average the value vectors $\mathbf{v}_j(R, h)$ over the group to
355 further collapse the geometric representation.

356 5 EXPERIMENTS
358

359 To validate our proposed architecture, we conduct a series of experiments across a number of different
360 tasks and datasets. Our evaluation is structured to analyze the role of the equivariance inductive bias
361 by categorizing tasks into two distinct settings based on their inherent geometric properties.

362 First, for tasks with inherent symmetry, such as those in QM9 (Ramakrishnan et al., 2014) and
363 OMol25 (Levine et al., 2025), the underlying molecular systems have no canonical orientation.
364 Their properties are determined by the relative positions of atoms and are independent of the global
365 coordinate system. Since the physical laws governing these molecular properties are $E(3)$ -symmetric,
366 equivariance becomes a fundamental requirement for a model to generalize efficiently (Fuchs et al.,
367 2020; Bronstein et al., 2021; Batzner et al., 2022; Pacini et al., 2025; Vadgama et al., 2025). We refer
368 to this category of problems as *Equivariant Tasks*.

369 Second, for tasks involving datasets with a canonical orientation, like CIFAR-10 (Krizhevsky,
370 2009) and ScanObjectNN (Uy et al., 2019), strict end-to-end equivariance is not required (the
371 images/objects are aligned w.r.t. a canonical up-direction). These problems nevertheless provide
372 a testbed to investigate if the geometric inductive bias of our model, enforced by weight-sharing,
373 improves performance on its own merits. We refer to these as *Non-Equivariant Tasks*.

374 375 5.1 EXPERIMENTAL SETUP
376

377 All Platonic Transformer variants are built upon RoPE, making them inherently *translation-equivariant*. The degree of rotational equivariance is then determined by the choice of a discrete

378 symmetry group $\mathcal{G} \subset O(n)$ that defines the set of reference frames. For instance, selecting the trivial
 379 group ($\mathcal{G} = \emptyset$) results in a purely translation-equivariant model ($T(n)$); it uses only the identity frame.
 380 Choosing the rotational symmetry group of the Tetrahedron provides 12 reference frames, making
 381 the model approximately $SE(n)$ -equivariant, or $E(n)$ -equivariant when including reflections too.
 382

383 For fair comparison, we match the computational cost between $SE(n)$ and $T(n)$ models by equating
 384 our group-based parallelism with standard multi-head attention. For instance, an $SE(n)$ model using
 385 the 12-element tetrahedral group with one head per frame is benchmarked against a $T(n)$ baseline
 386 with 12 total heads (details in App. G). For certain tasks, symmetries can be conditionally broken by
 387 using APE or providing an *external reference frame*, yet internal layers critically retain principled
 388 weight-sharing. Using external frames to break symmetry and APE to provide geometric information
 389 are effective strategies, allowing a model to benefit from geometric inputs without being end-to-end
 390 constrained by full equivariance (Vadgama et al., 2025).

391 For CIFAR-10 and ScanObjectNN, we conducted a comprehensive sweep to find the optimal configura-
 392 tion. In contrast, for QM9 and OMol25, we used a sequential process: first, we identified the best
 393 architecture via an extensive sweep on QM9, then transferred these hyperparameters to OMol25 for
 394 further refinement with a one-million subset before full training (see Appendix H-K).
 395
 396

5.2 NON-EQUIVARIANT TASKS

397 **Cifar10** The results of our ablation study on
 398 CIFAR-10 are presented in Table 1. The findings
 399 indicate that incorporating 2D rotational symmetries
 400 provides a tangible benefit over the translation-only
 401 baseline (the \emptyset^* model, which is equivalent to a stan-
 402 dard Vision Transformer). This suggests that even for
 403 general-purpose vision tasks without an end-to-end
 404 equivariance requirement, equivariance proves to be
 405 an important inductive bias. This may be explained
 406 by the fact that even though images have a canonical
 407 pose (e.g. with the sky at the top), equivariance al-
 408 lows for internal weight-sharing and thus the reuse of
 409 patterns (edges, parts, objects) that may appear at ar-
 410bitrary orientations within an image. The comparison
 411 between the full attention and linear-convolutional
 412 shows a significant impact of attention over the lin-
 413 ear complexity dynamic convolution counterpart (in
 414 which the softmax is omitted, cf. Prop. 2).
 415
 416

417 **ScanObjectNN** On the ScanObjectNN point cloud
 418 classification task, we test the effectiveness of 3D
 419 symmetry groups (trivial vs tetrahedron vs horizontal
 420 flips) in a realistic setting with occlusions and signif-
 421 icant orientation variability. The results, shown in
 422 Table 2 again highlight the impact of equivariance
 423 and weightsharing. While the quadratic-cost atten-
 424 tion mechanism offers greater expressive power, the
 425 linear-time convolutional variant provides a signif-
 426 icant speed-up, which is critical for efficiently pro-
 427 cessing large point clouds. This demonstrates the
 428 versatility of our approach in adapting to different
 429 computational and modeling requirements in 3D com-
 430 puter vision, as demonstrated in Figure 3. Also note
 431 that the computational cost is independent of the cho-
 sen symmetry group.

Table 1: CIFAR-10 Accuracy (%).

Group	Attention	Conv	# Params
	Acc. (\uparrow)	Acc. (\uparrow)	
\emptyset^*	91.49	88.56	85.1M
C_4	92.73	88.70	21.3M
C_6	92.16	88.44	14.2M
D_4	92.53	88.10	21.3M
D_6	92.07	88.58	14.2M
Flop	91.49	87.86	42.6M

Table 2: ScanObjectNN Overall Acc. (%).

Group	Attention	Conv
	Acc. (\uparrow)	Acc. (\uparrow)
\emptyset^*	80.5	79.8
Tetrahedron	81.3	80.1
Flop	82.0	80.6

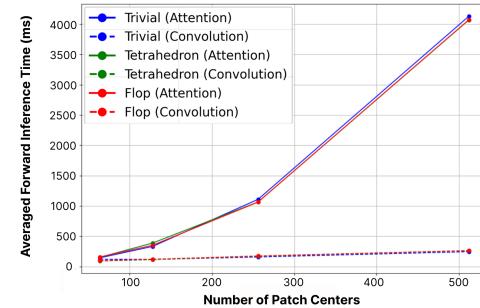


Figure 3: The Platonic Transformer, when configured in its convolutional mode, exhibits a linear computational complexity relative to the input sequence length, a complexity shared with its attention mode. It is noteworthy that the model’s specific equivariance type does not alter this computational scaling.

432

Table 3: QM9 Property Prediction MAE (↓).

Platonic Transformer (end-to-end)					
Group	Attention		Convolution		
	μ	α	μ	α	
\emptyset	0.028	0.064	0.030	0.061	
Tetrahedron	0.012	0.049	0.014	0.047	
Octahedron	0.010	0.048	0.012	0.047	

Platonic Transformer with PCA-based frames					
Method	8-Refl + 1 frame	0.039	0.155	–	–

Reference methods					
Method	*(re)produced in this work				
EquiformerV2 [38]	0.010	0.050	–	–	
FAFormer [28]*	0.122	0.252	–	–	
G-Hyena [42]*	–	–	0.018	0.066	
Rapidash [62]	–	–	0.010	0.040	

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

5.3 EQUIVARIANT TASKS

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

QM9 Our results on the QM9 benchmark are summarized in Table 7. The first group of results identifies the most effective Platonic group and model variant for this task, showing a performance gain from incorporating Platonic symmetries over the translation-only (\emptyset group) baseline. Both the Tetrahedron and Octahedron groups achieve strong performance, delivering results on par with state-of-the-art methods like EquiformerV2 (Liao et al., 2024). Our linear-convolutional models are similarly effective, outperforming other convolution-based baselines such as G-Hyena (Moskalev et al., 2025), a state-space model for long-context geometric modeling. Our end-to-end $SE(3)$ -equivariant model is superior to existing baselines, demonstrating that a directly learned geometric representation is fundamentally more effective compared to external symmetrization.

Our use of reference frames relates to the popular class of “frame-based methods,” which are a popular approach for building equivariant networks Du et al. (2023); Yin et al. (2025); Puny et al. (2022). One such prominent framework is Frame Averaging (FA), which incorporates group equivariance via *symmetrization* (detailed introduction in Appendix E) of non-equivariant neural network backbones. FA can be extended to full $E(3)$ by including roto-reflections (e.g., roto-reflecting PCA axes) but comes at a cost, requiring a separate forward pass for each frame element. Our Platonic Transformer offers a more scalable drop-in replacement. For instance, using just a single orthogonal matrix from PCA (i.e., just one frame element) along with an axis-aligned roto-reflection group (here, we use $C_2 \times C_2 \times C_2$, a subgroup of the octahedral group) achieves native $E(3)$ equivariance at an ~ 8 times lower cost. We evaluate the effectiveness of FA by comparing to FAFormer (Huang et al., 2024), which applies FA to a standard Transformer backbone with 8 frame elements. Our single-frame variant with only octahedral reflections (8-Refl + 1 frame) outperforms FAFormer with a smaller compute cost. This underscores the need for a geometrically expressive backbone, such as the Platonic Transformer.

475

476

477

478

479

480

481

482

483

484

485

OMol25 To validate the scalability and performance of our proposed architecture, we evaluate our model with the best hyperparameters on the large-scale OMol25 dataset. For a fair comparison under a constrained computational budget, we compare the Platonic Transformer with eSEN (Levine et al., 2025), the current state-of-the-art method on OMol25, for 120 hours on a node with 4 NVIDIA 6000Ada GPUs; note that we use the hyperparameters for eSEN indicated in the original work. The results are presented in Table 5, demonstrating that our model significantly outperforms the baseline under these identical conditions for energy prediction and achieves highly competitive performance in force prediction. This performance is noteworthy when contextualized against current literature benchmarks, which we estimate from Levine et al. (2025), utilized a computational budget nearly four times larger. Achieving strong results under such constraints indicates the architectural efficiency of our model, suggesting that the Platonic Transformer could likely achieve state-of-the-art accuracy with a comparable computational budget which we leave for future work.

Table 4: Inference wall-clock times on QM9.

Platonic Transformer		Avg. Time (ms) (↓)
Group		
\emptyset		2.87 ± 0.29
Tetrahedron		2.79 ± 0.21
Octahedron		2.85 ± 0.25

Reference methods		
Method		
Standard Transformer		2.01 ± 3.74
G-Hyena [42]		44.06 ± 60.05
TFN [58]		590.45 ± 269.25

Table 5: OMol25 Energy/Force MAE (↓).			
Our own (re)produced results, (4 GPUs - 120 hrs)			
Method	Force	Energy	E/Atom
Platonic Transformer	24.25	74.00	2.63
eSEN [36]	23.92	120.0	3.37

From literature, (estimated 4 GPUs - 475 hrs)			
eSEN [36]	10.11	29.80	0.88
MACE [36]	16.83	54.09	1.55

486 **Inference times** Given that the Platonic Transformer does not alter the computation graph of the
 487 standard Transformer, our method benefits from similarly fast inference speeds. A single Platonic
 488 Transformer layer runs at the same order of magnitude as a standard Transformer layer (implemented
 489 using a single `TransformerEncoderLayer` module in PyTorch) on a batch size of 64 molecules
 490 from QM9 on a single H200 GPU, averaged over 10 batches, as shown in Table 4. We also show
 491 superior inference times, by 2-3 orders of magnitude, against a single G-Hyena and Tensor Field
 492 Network (Thomas et al., 2018) layer under the same setup while still retaining $E(3)$ equivariance.
 493

494 6 CONCLUSION

495
 496 We introduce the Platonic Transformer, a framework that achieves approximate $E(n)$ equivariance
 497 without compromising the flexibility and scalability of the standard Transformer architecture. By
 498 combining Rotary Position Embeddings (RoPE) with a new frame-dependent attention mechanism—
 499 where attention is computed relative to reference frames from Platonic solid symmetry groups—we
 500 integrate a powerful geometric inductive bias while preserving the original computation graph and
 501 cost. This approach demonstrates that principled equivariance and modern scalability are not mutually
 502 exclusive. Furthermore, our analysis reveals a formal equivalence to dynamic group convolution
 503 with linear complexity, enabling a highly scalable, linear-time variant for large-scale tasks. In many
 504 scientific domains, equivariance represents a "Platonic ideal" — an essential physical principle a
 505 model should respect. By eliminating the trade-off between this principled design and computational
 506 efficiency, the Platonic Transformer makes this ideal a practical and scalable reality.
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539

540 REPRODUCIBILITY STATEMENT
541

542 We make several strides towards reproducibility of our work. We back our theoretical results with
543 proofs in Appendix C, and with rough intuitions in the main text. We have attached a `.zip` folder as
544 supplementary material with source code to train our Platonic Transformer models (and baselines)
545 across different tasks and datasets; we also provide adequate comments and documentation. Details
546 required to reproduce the results in our tables and figures are provided in Appendices H-K. We
547 also provide information on hyperparameter-tuning efforts in Appendices F and G. We intend to
548 open-source the code on public platforms like GitHub once the review period has formally ended.
549

550 ETHICS STATEMENT
551

552 Our work addresses fundamental challenges in building scalable methods for deep learning and
553 AI for Science. We believe there is potential for our methods to be used in scientific domains like
554 biotechnology as well as energy and sustainability. As our work is in its early stages, we posit that it
555 introduces dedicated computational methods rather than focusing on particular applications that may
556 warrant closer oversight and caution.
557

558 REFERENCES
559

560 Serge Assaad, Carlton Downey, Rami Al-Rfou', Nigamaa Nayakanti, and Benjamin Sapp. VN-
561 transformer: Rotation-equivariant attention for vector neurons. *Transactions on Machine Learn-
562 ing Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=EiX2L4sDPG>.
563

564 Julia Balla, Siddharth Mishra-Sharma, Carolina Cuesta-Lazaro, Tommi Jaakkola, and Tess Smidt.
565 A cosmic-scale benchmark for symmetry-preserving data processing, 2024. URL <https://arxiv.org/abs/2410.20516>.
566

567 Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Ko-
568 rnbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural
569 networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13
570 (1):2453, 2022. doi: 10.1038/s41467-022-29939-5. URL <https://www.nature.com/articles/s41467-022-29939-5>.
571

572 Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*,
573 2020. URL <https://openreview.net/forum?id=H1gBhkBFDH>.
574

575 Erik J Bekkers, Sharvaree Vadgama, Rob Hesselink, Putri A Van der Linden, and David W. Romero.
576 Fast, expressive $\mathcal{SE}(n)$ equivariant networks through weight-sharing in position-
577 orientation space. In *The Twelfth International Conference on Learning Representations*, 2024.
578 URL <https://openreview.net/forum?id=dPHLbUqGbr>.
579

580 Georg Bökman, David Nordström, and Fredrik Kahl. Flopping for flops: Leveraging equivariance for
581 computational efficiency. In *Forty-second International Conference on Machine Learning*, 2025.
582

583 Avishhek Joey Bose, Tara Akhoud-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks,
584 Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander
585 Tong. Se(3)-stochastic flow matching for protein backbone generation, 2024. URL <https://arxiv.org/abs/2310.02391>.
586

587 Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Ge-
588 ometric and physical quantities improve $e(3)$ equivariant message passing. In *International
589 Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_xwr8gOBeV1.
590

591 Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco Cohen. Geometric algebra transformer,
592 2023. URL <https://arxiv.org/abs/2305.18415>.
593

594 Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning:
 595 Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. URL
 596 <https://arxiv.org/abs/2104.13478>.

597 Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build e (n)-equivariant steerable cnns.
 598 In *International conference on learning representations*, 2022.

599

600 Nan Chen and Soledad Villar. Se (3)-equivariant self-attention via invariant features. In *Machine*
 601 *Learning for Physics NeurIPS Workshop*, 2022.

602

603 Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of
 604 large language models via positional interpolation, 2023.

605 Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas
 606 Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention
 607 with performers. *arXiv preprint arXiv:2009.14794*, 2020.

608

609 Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference*
 610 *on machine learning*, pp. 2990–2999. PMLR, 2016.

611

612 Taco S. Cohen and Max Welling. Steerable cnns. In *International Conference on Learning Representations (ICLR)*, 2017.

613

614 Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homoge-
 615 neous spaces. *Advances in neural information processing systems*, 32, 2019.

616

617 Zihang Dai, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan
 618 Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

619

620 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep
 621 bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of*
 622 *the North American Chapter of the Association for Computational Linguistics: Human Language*
 623 *Technologies*, pp. 4171–4186, 2019.

624

625 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas
 626 Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit,
 627 and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale.
 628 In *International Conference on Learning Representations (ICLR)*, 2021.

629

630 Yuanqi Du, Limei Wang, Dieqiao Feng, Guifeng Wang, Shuiwang Ji, Carla P Gomes, Zhi-Ming Ma,
 631 et al. A new perspective on building efficient and expressive 3d equivariant graph neural networks.
 632 *Advances in neural information processing systems*, 36:66647–66674, 2023.

633

634 J Thorben Frank, Stefan Chmiela, Klaus-Robert MĂžller, and Oliver T Unke. Euclidean fast attention:
 635 Machine learning global atomic representations at linear cost. *arXiv preprint arXiv:2412.08541*,
 636 2024.

637

638 Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d
 639 roto-translation equivariant attention networks. In *Advances in Neural Information Processing*
 640 *Systems (NeurIPS)*, 2020.

641

642 Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan GĂ¶nnemann. Fast and
 643 uncertainty-aware directional message passing for non-equilibrium molecules. In *NeurIPS 2020*
 644 *Workshop on Machine Learning for Molecules*, 2020. URL <https://arxiv.org/abs/2011.14115>.

645

646 Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural
 647 message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings*
 648 *of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine*
 649 *Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/gilmer17a.html>.

650

651 Lingshen He, Yuxuan Chen, Zhengyang Shen, Yiming Dong, Yisen Wang, and Zhouchen Lin.
 652 Efficient equivariant network. In *Advances in Neural Information Processing Systems*, 2021a.

648 Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert
 649 with disentangled attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021b. URL <https://openreview.net/forum?id=XPZIaotutsD>.
 650

651 Tinglin Huang, Zhenqiao Song, Rex Ying, and Wengong Jin. Protein-nucleic acid complex modeling
 652 with frame averaging transformer, 2024. URL <https://arxiv.org/abs/2406.09586>.
 653

654 Chaitanya K. Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop
 655 Sriram, and Zachary W. Ulissi. All-atom diffusion transformers: Unified generative modelling of
 656 molecules and materials, 2025. URL <https://arxiv.org/abs/2503.03965>.
 657

658 John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger,
 659 Kathryn Tunyasuvunakool, Russell Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate
 660 protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
 661

662 David M Knigge, David R Wessels, Riccardo Valperga, Samuele Papa, Jan-Jakob Sonke, Efstratios
 663 Gavves, and Erik J Bekkers. Space-time continuous pde forecasting using equivariant neural fields.
 664 *Advances in Neural Information Processing Systems*, 37:76553–76577, 2024.
 665

666 Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical re-
 667 port, University of Toronto, 2009. URL <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
 668

669 Soumyabrata Kundu and Risi Kondor. Steerable transformers for volumetric data. In *Forty-second
 670 International Conference on Machine Learning*, 2025.
 671

672 Hannah Lawrence, Vasco Portilheiro, Yan Zhang, and Sékou-Oumar Kaba. Improving equivariant
 673 networks with probabilistic symmetry breaking, 2025. URL <https://arxiv.org/abs/2503.21985>.
 674

675 Tuan Le, Frank Noé, and Djork-Arné Clevert. Equivariant graph attention networks for molecular
 676 property prediction. *arXiv preprint arXiv:2202.09891*, 2022.
 677

678 Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, Muham-
 679 mad R. Hasim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman,
 680 Nathan C. Frey, Xiang Fu, Vahe Gharakhanyan, Aditi S. Krishnapriyan, Joshua A. Rackers, San-
 681jeev Raja, Ammar Rizvi, Andrew S. Rosen, Zachary Ulissi, Santiago Vargas, C. Lawrence Zitnick,
 682 Samuel M. Blau, and Brandon M. Wood. The open molecules 2025 (omol25) dataset, evaluations,
 683 and models, 2025. URL <https://arxiv.org/abs/2505.08762>.
 684

685 Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic
 686 graphs. In *The Eleventh International Conference on Learning Representations*, 2023. URL
 687 <https://openreview.net/forum?id=KwmPfARgOTD>.
 688

689 Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved equivariant
 690 transformer for scaling to higher-degree representations, 2024. URL <https://arxiv.org/abs/2306.12059>.
 691

692 Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical
 693 message passing for 3d molecular graphs. In *International Conference on Learning Representations*,
 694 2022. URL <https://openreview.net/forum?id=givsRXs0t9r>.
 695

696 Shengjie Luo, Tianlang Chen, and Aditi S. Krishnapriyan. Enabling efficient equivariant opera-
 697 tions in the fourier basis via gaunt tensor products. In *International Conference on Learning
 698 Representations*, 2024. Spotlight.
 699

700 Dominic Masters, Josef Dean, Kerstin Klaser, Zhiyi Li, Sam Maddrell-Mander, Adam Sanders,
 701 Hatem Helal, Deniz Beker, Ladislav Rampášek, and Dominique Beaini. Gps++: An optimised
 702 hybrid mpnn/transformer for molecular property prediction. *arXiv preprint arXiv:2212.02229*,
 703 2022.
 704

705 Artem Moskalev, Mangal Prakash, Junjie Xu, Tianyu Cui, Rui Liao, and Tommaso Mansi. Geometric
 706 hyena networks for large-scale equivariant learning, 2025. URL <https://arxiv.org/abs/2505.22560>.
 707

702 David Nordström, Johan Edstedt, Fredrik Kahl, and Georg Bökman. Stronger vits with octic
 703 equivariance. *arXiv preprint arXiv:2505.15441*, 2025.

704

705 Elias Nyholm, Oscar Carlsson, Maurice Weiler, and Daniel Persson. Equivariant non-linear maps
 706 for neural networks on homogeneous spaces, 2025. URL <https://arxiv.org/abs/2504.20974>.

707

708 Marco Pacini, Gabriele Santin, Bruno Lepri, and Shubhendu Trivedi. On universality classes of
 709 equivariant networks, 2025. URL <https://arxiv.org/abs/2506.02293>.

710

711 Yatian Pang, Eng Hock Francis Tay, Li Yuan, and Zhenghua Chen. Masked autoencoders for 3d
 712 point cloud self-supervised learning. *World Scientific Annual Review of Artificial Intelligence*, 1:2440001, 2023.

713

714 Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J. Smith, and
 715 Yaron Lipman. Frame averaging for invariant and equivariant network design, 2022. URL
<https://arxiv.org/abs/2110.03336>.

716

717

718 Eric Qu and Aditi S. Krishnapriyan. The importance of being scalable: Improving the speed
 719 and accuracy of neural network interatomic potentials across chemical domains, 2024. URL
<https://arxiv.org/abs/2410.24169>.

720

721

722 Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum
 723 chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014.

724

725 B Srinivasa Reddy and Biswanath N Chatterji. An fft-based technique for translation, rotation, and
 726 scale-invariant image registration. *IEEE transactions on image processing*, 5(8):1266–1271, 1996.

727

728 David W. Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention
 729 for vision. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=JkfYjnOEo6M>.

730

731 Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks.
 732 In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

733

734 Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction
 735 of tensorial properties and molecular spectra. In *International conference on machine learning*, pp.
 9377–9388. PMLR, 2021.

736

737 Jean-Pierre Serre. *Linear representations of finite groups*, volume 42. Springer, 1977.

738

739 Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations.
 740 In *Proceedings of NAACL-HLT 2018 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, 2018.
 doi: 10.18653/v1/N18-2074. URL <https://aclanthology.org/N18-2074/>.

741

742 Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced
 743 transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

744

745 Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular
 746 potentials. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=zNHzqZ9wrRB>.

747

748 Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley.
 749 Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds,
 750 2018. URL <https://arxiv.org/abs/1802.08219>.

751

752 Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European conference
 753 on computer vision*, pp. 516–533. Springer, 2022.

754

755 Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das,
 Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The open catalyst 2022
 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis*, 13(5):3066–3084, 2023.

756 Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung.
 757 Revisiting point cloud classification: A new benchmark dataset and classification model on
 758 real-world data. In *International Conference on Computer Vision (ICCV)*, 2019.

759

760 Sharvaree Vadgama, Mohammad Mohaiminul Islam, Domas Buracas, Christian Shewmake, Artem
 761 Moskalev, and Erik Bekkers. Probing equivariance and symmetry breaking in convolutional
 762 networks, 2025. URL <https://arxiv.org/abs/2501.01999>.

763

764 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz
 765 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing
 766 systems*, 30, 2017.

767

768 David R Wessels, David M Knigge, Samuele Papa, Riccardo Valperga, Sharvaree Vadgama, Efstratios
 769 Gavves, and Erik J Bekkers. Grounding continuous representations in geometry: Equivariant
 770 neural fields. *arXiv preprint arXiv:2406.05753*, 2024.

771

772 Bangchen Yin, Jiaao Wang, Weitao Du, Pengbo Wang, Penghua Ying, Haojun Jia, Zisheng Zhang,
 773 Yuanqi Du, Carla P Gomes, Chenru Duan, et al. Alphanet: Scaling up local frame-based atomistic
 774 foundation model. *arXiv e-prints*, pp. arXiv–2501, 2025.

775

776 Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and
 777 Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in
 778 Neural Information Processing Systems (NeurIPS)*, 2021.

779

780 Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-
 781 training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF
 782 conference on computer vision and pattern recognition*, pp. 19313–19322, 2022.

783

784 Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In
 785 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16259–
 786 16268, 2021.

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

Appendix

Table of Contents

A	Rotary Position Embeddings from a Group Theoretical Perspective	18
A.1	The Theoretical Toolkit	18
A.2	Constructing the RoPE Operator	18
A.3	Translation Invariance in Attention	19
A.4	A Fourier Perspective	19
B	Equivariance Properties of Platonic Transformers	19
C	Proofs	20
C.1	Proof of Proposition 2	20
C.2	Proof of Corollary 1	21
C.3	Proof of Platonic Transformers Implementing Group Convolutions	21
D	Equivalent Attention via RoPE Base Frequency Steering	21
E	Frame Averaging	22
F	Details of Architecture	22
G	Hyperparameter Tuning and Model Selection Strategy	23
H	Details of experiments on Cifar10	24
H.1	Description of the dataset	24
H.2	Training Details	24
I	Details of experiments on ScanObjectNN	24
I.1	Description of the dataset	24
I.2	Training Details	24
J	Details of experiments on QM9	25
J.1	Description of the dataset	25
J.2	Training Details for the Regression Experiment	25
J.3	Additional details on wall-clock timings	25
J.4	Extended experiments on QM9	25
K	Details of experiments on OMol25	26
K.1	Description of the dataset	26
K.2	Training Details	27
L	Experiments on Learned Key Projections	27
L.1	Instability of Learned Keys	27
L.2	Mitigating Instability via Regularization	28
M	Visualizations of Learned Attention Scores	29
N	Implementing Platonic Transformers in the Fourier Domain of Finite Groups	29
N.1	Introduction to the Fourier Theory of Finite Groups	30
N.2	Fourier Theory of the Tetrahedral Group	30
N.3	Implementation	31

864	N.4 Throughput Benchmarking	32
865		
866	O Disclosure of LLM Usage	32
867		
868		
869		
870		
871		
872		
873		
874		
875		
876		
877		
878		
879		
880		
881		
882		
883		
884		
885		
886		
887		
888		
889		
890		
891		
892		
893		
894		
895		
896		
897		
898		
899		
900		
901		
902		
903		
904		
905		
906		
907		
908		
909		
910		
911		
912		
913		
914		
915		
916		
917		

918 **A ROTARY POSITION EMBEDDINGS FROM A GROUP THEORETICAL
919 PERSPECTIVE**

921 A fundamental challenge in geometric deep learning is creating position representations that respect
922 underlying symmetries. For data in \mathbb{R}^d , our goal is to define a high-dimensional position embedding,
923 $\mathbf{E} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, that is equivariant to translations. This requires that for any translation vector \mathbf{p} , the
924 embedding transforms predictably: $\mathbf{E}(\mathbf{p}_0 + \mathbf{p}) = \rho(\mathbf{p})\mathbf{E}(\mathbf{p}_0)$, where $\rho(\mathbf{p})$ is a linear transformation.
925 Group representation theory provides the formal tools to construct such embeddings.
926

927 **A.1 THE THEORETICAL TOOLKIT**

929 To proceed, we first define the essential concepts required for our construction.

930 **Definition 1** (Representation). *A linear representation of a group \mathcal{G} on a vector space V is a group
931 homomorphism $\rho : \mathcal{G} \rightarrow GL(V)$, where $GL(V)$ is the general linear group of invertible linear
932 transformations on V .*

934 To ensure that the positional encoding does not arbitrarily amplify or diminish feature magnitudes,
935 which would destabilize learning, we require the representations to be length-preserving. This leads
936 to the concept of a unitary representation.

937 **Definition 2** (Unitary Representation). *A representation ρ is **unitary** if it maps group elements
938 to unitary operators, i.e., $\rho : \mathcal{G} \rightarrow U(V)$. For real-valued representations, this corresponds to
939 orthogonality, $\rho(g)^{-1} = \rho(g)^\top$.*

940 Just as a complex signal can be decomposed into pure frequencies, a general representation can be
941 broken down into fundamental building blocks known as irreducible representations (irreps).

943 **Definition 3** (Irreducible Representation). *An **irreducible representation** (irrep) is a representation
944 acting on a vector space V that has no non-trivial invariant subspaces.*

946 **A.2 CONSTRUCTING THE ROPE OPERATOR**

948 With these formal tools, we can now build the RoPE operator. The irreps of the translation group
949 $(\mathbb{R}^d, +)$ are indexed by a frequency vector $\omega \in \mathbb{R}^d$ and are given by one-dimensional, unitary
950 representations:

$$\rho_\omega(\mathbf{p}) = e^{i\omega^\top \mathbf{p}}. \quad (13)$$

952 This exponential form is the unique continuous solution to the group's homomorphism property,
953 $\rho(\mathbf{p}_1 + \mathbf{p}_2) = \rho(\mathbf{p}_1)\rho(\mathbf{p}_2)$, where the imaginary exponent ensures unitarity.

954 However, neural networks typically operate on real numbers. We can obtain a real-valued irrep by
955 combining pairs of conjugate frequencies, ω_k and $-\omega_k$. This yields a 2D irreducible representation
956 that takes the familiar form of a rotation matrix:

$$\rho_{\omega_k}(\mathbf{p}) = \begin{pmatrix} \cos(\omega_k^\top \mathbf{p}) & -\sin(\omega_k^\top \mathbf{p}) \\ \sin(\omega_k^\top \mathbf{p}) & \cos(\omega_k^\top \mathbf{p}) \end{pmatrix}. \quad (14)$$

960 To create a high-dimensional embedding, we simply select a set of frequencies $\Omega = \{\omega_k\}_{k=1}^{d'/2}$
961 and stack these 2D rotation blocks along the diagonal of a larger matrix. This results in a single,
962 block-diagonal transformation that correctly and equivariantly updates the entire embedding for a
963 given translation \mathbf{p} :

$$\rho_\Omega(\mathbf{p}) = \text{diag}(\rho^{\omega_1}(\mathbf{p}), \dots, \rho_{\omega_{d'/2}}(\mathbf{p})). \quad (15)$$

966 This is the core mechanism behind Rotary Position Embeddings. Its structure guarantees both
967 equivariance and computational efficiency, as each 2D component can be rotated independently.

968 **Definition 4** (Rotary Position Embedding (RoPE) Operator). *The RoPE operator $\rho_\Omega(\mathbf{p})$ for a
969 position $\mathbf{p} \in \mathbb{R}^d$ is the block-diagonal rotation matrix defined above (Equation 15), constructed
970 from a set of frequencies Ω . The application of RoPE to a feature vector $\mathbf{f} \in \mathbb{R}^{d'}$ is defined as the
971 matrix-vector product: $\rho_\Omega(\mathbf{p})\mathbf{f}$. For this operation to be well-defined, the feature dimension d' must
be even.*

972 A.3 TRANSLATION INVARIANCE IN ATTENTION
973974 While the RoPE operator provides an *equivariant* transformation for feature vectors, its crucial benefit
975 within the Transformer architecture is that it makes the attention score *invariant* to global translations.
976 This property ensures that the attention mechanism only considers the relative positions of tokens,
977 which is the inductive bias we seek. We formalize this key result below.978 **Proposition 3** (Translation Invariance of the RoPE Attention Score). *The attention score computed
979 using RoPE, $\text{attn}(\mathbf{q}, \mathbf{k}, \Delta\mathbf{p}) = \mathbf{q}^\top \rho_\Omega(\Delta\mathbf{p}) \mathbf{k}$, where $\Delta\mathbf{p} = \mathbf{p}_j - \mathbf{p}_i$, is invariant to a global
980 translation of the coordinate system.*981
982 *Proof.* Let the positions \mathbf{p}_i and \mathbf{p}_j be translated by an arbitrary vector $\mathbf{t} \in \mathbb{R}^d$, resulting in new
983 positions $\mathbf{p}'_i = \mathbf{p}_i + \mathbf{t}$ and $\mathbf{p}'_j = \mathbf{p}_j + \mathbf{t}$. The new relative displacement vector, $\Delta\mathbf{p}'$, is:

984
985
$$\Delta\mathbf{p}' = \mathbf{p}'_j - \mathbf{p}'_i = (\mathbf{p}_j + \mathbf{t}) - (\mathbf{p}_i + \mathbf{t}) = \mathbf{p}_j - \mathbf{p}_i = \Delta\mathbf{p}. \quad (16)$$

986 Since the relative displacement vector is unchanged by the global translation, the RoPE operator
987 applied to it also remains unchanged: $\rho_\Omega(\Delta\mathbf{p}') = \rho_\Omega(\Delta\mathbf{p})$. Consequently, the attention score, which
988 depends only on the content vectors and this operator, is invariant to the translation:
989

990
991
$$\mathbf{q}^\top \rho_\Omega(\Delta\mathbf{p}') \mathbf{k} = \mathbf{q}^\top \rho_\Omega(\Delta\mathbf{p}) \mathbf{k}. \quad (17)$$

992 This formally demonstrates that RoPE imparts translation invariance to the attention mechanism. \square
993994 A.4 A FOURIER PERSPECTIVE
995996 The principle of constructing equivariant functions from irreducible representations is deeply connected
997 to Fourier analysis. The Fourier transform provides a way to decompose any function on a
998 group into a weighted sum (or integral) over its irreps. For the translation group on \mathbb{R}^d , these irreps
999 are precisely the complex exponentials we used as our building blocks. Therefore, RoPE can be
1000 understood as a practical application of Fourier theory, using a discrete basis of Fourier modes (the
chosen frequencies Ω) to represent the positional signal.1001 **Definition 5** (Fourier Transform on \mathbb{R}^d). *The forward Fourier transform $\mathcal{F} : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$
1002 maps a function f to its frequency-space representation \hat{f} . The coefficient for a frequency ω is the
1003 projection of f onto the corresponding irrep ρ_ω :*

1004
1005
$$\hat{f}(\omega) = \mathcal{F}\{f\}(\omega) = \int_{\mathbb{R}^d} f(\mathbf{p}) \overline{\rho_\omega(\mathbf{p})} d\mathbf{p} = \int_{\mathbb{R}^d} f(\mathbf{p}) e^{-i\omega^\top \mathbf{p}} d\mathbf{p}. \quad (18)$$

1006 The inverse transform reconstructs the function by integrating over all irreps:
1007

1008
1009
$$f(\mathbf{p}) = \mathcal{F}^{-1}\{\hat{f}\}(\mathbf{p}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) \rho_\omega(\mathbf{p}) d\omega = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top \mathbf{p}} d\omega. \quad (19)$$

1010 B EQUIVARIANCE PROPERTIES OF PLATONIC TRANSFORMERS
10111012 We formally establish the equivariance of our proposed architecture. We consider a point cloud
1013 $\{\mathbf{p}_i, \mathbf{v}_i, s_i\}_{i=1}^N$ consisting of positions, vectors, and scalars. A global rotation $R \in \mathcal{G}$ acts on these
1014 inputs as $\mathbf{p}_i \mapsto R\mathbf{p}_i$, $\mathbf{v}_i \mapsto R\mathbf{v}_i$, and $s_i \mapsto s_i$.1015 **Equivariant Feature Lifting.** Input features are first lifted to functions on the group \mathcal{G} . The
1016 lifting operator, Lift , maps the input point cloud to a set of feature maps $\{\mathbf{f}_i : \mathcal{G} \rightarrow \mathbb{R}^C\}_{i=1}^N$. Scalar
1017 components are copied to each frame, while vector components (from $\mathbf{p}_i, \mathbf{v}_i$) are lifted by projecting
1018 them onto each reference frame. This projection means expressing the vector's coordinates in the
1019 local basis of a given frame $\tilde{R} \in \mathcal{G}$, which is achieved by the transformation $\tilde{R}^{-1}\mathbf{v}$. This lifting
1020 procedure is equivariant by construction: a global rotation R of the input point cloud results in the
1021 lifted feature maps transforming via the left regular representation, L_R . That is:
1022

1023
1024
$$(\text{Lift}(R \cdot \text{cloud}))_i(\tilde{R}) = (\text{Lift}(\text{cloud}))_i(R^{-1}\tilde{R}) \triangleq (L_R \mathbf{f}_i)(\tilde{R}). \quad (20)$$

1026 **Equivariant Linear Layers.** All linear layers Φ in our network are implemented as point-wise
 1027 group convolutions, as shown in Eq. 8. These layers are equivariant to the action of the group by
 1028 construction (Cohen et al., 2019, Thm. 3.1), satisfying $\Phi(L_R \mathbf{f}_i) = L_R(\Phi(\mathbf{f}_i))$.
 1029

1030 This leads to our main proposition regarding the attention mechanism.

1031 **Proposition 4** (Equivariant Attention). *Let the queries Q_i , keys K_i , and values V_i be equivariant
 1032 feature maps produced by the equivariant linear layers. The RoPE-enhanced attention mechanism
 1033 (Eq. 7), which computes outputs \mathbf{y}_i , is an equivariant operation. That is, if the inputs transform as
 1034 $\mathbf{f}_i \mapsto L_R \mathbf{f}_i$, the outputs transform as $\mathbf{y}_i \mapsto L_R \mathbf{y}_i$.*

1035 *Proof.* Let the inputs to the attention layer (Q_i, K_i, V_i) transform under a global rotation R as
 1036 $Q'_i = L_R Q_i$, $K'_i = L_R K_i$, and $V'_i = L_R V_i$. We analyze the transformation of each component of
 1037 the attention calculation.

1038 The score function $s_{ij}(\tilde{R})$, which depends on $Q_i(\tilde{R})$ and $K_j(\tilde{R})$ (and potentially RoPE terms derived
 1039 from lifted positions), will transform as:

$$1041 \quad s'_{ij}(\tilde{R}) = \text{score}(Q'_i(\tilde{R}), K'_i(\tilde{R}), \dots) = \text{score}(Q_i(R^{-1}\tilde{R}), K_j(R^{-1}\tilde{R}), \dots) = s_{ij}(R^{-1}\tilde{R}).$$

1043 This means the score function itself is equivariant, $s'_{ij} = L_R s_{ij}$. Since the softmax operator is applied
 1044 point-wise for each frame \tilde{R} over the index j , the attention weights also transform equivariantly:

$$1045 \quad \text{attn}'_{ij}(\tilde{R}) = \underset{j}{\text{softmax}}(s'_{ij}(\tilde{R})) = \underset{j}{\text{softmax}}(s_{ij}(R^{-1}\tilde{R})) = \text{attn}_{ij}(R^{-1}\tilde{R}).$$

1047 Finally, the output feature map \mathbf{y}_i transforms as:

$$1049 \quad \mathbf{y}'_i(\tilde{R}) = \sum_{j=1}^N \text{attn}'_{ij}(\tilde{R}) V'_j(\tilde{R})$$

$$1050 \quad = \sum_{j=1}^N \text{attn}_{ij}(R^{-1}\tilde{R}) V_j(R^{-1}\tilde{R}) = \mathbf{y}_i(R^{-1}\tilde{R}).$$

1055 Thus, the output transforms as $\mathbf{y}'_i = L_R \mathbf{y}_i$, proving the attention mechanism is equivariant. \square

1057 C PROOFS

1059 C.1 PROOF OF PROPOSITION 2

1061 We seek to show that the unnormalized attention score, which defines the kernel $\phi_{\mathbf{q}_i}(\Delta \mathbf{p})$, takes the
 1062 form of a sparse Fourier series whose coefficients are linear projections of the query \mathbf{q}_i .
 1063

1064 Let the relative position be $\Delta \mathbf{p} = \mathbf{p}_j - \mathbf{p}_i$. With a constant key vector $\mathbf{k}_j = \mathbf{1}$, the kernel is defined
 1065 by the attention score:

$$1066 \quad \phi_{\mathbf{q}_i}(\Delta \mathbf{p}) = (\rho(\mathbf{p}_i) \mathbf{q}_i)^\top (\rho(\mathbf{p}_j) \mathbf{1})$$

1067 Using the properties of the RoPE operator ρ , this simplifies to:

$$1068 \quad \phi_{\mathbf{q}_i}(\Delta \mathbf{p}) = \mathbf{q}_i^\top \rho(\mathbf{p}_i)^\top \rho(\mathbf{p}_j) \mathbf{1} = \mathbf{q}_i^\top \rho(\Delta \mathbf{p}) \mathbf{1}$$

1070 The RoPE matrix $\rho(\Delta \mathbf{p})$ is block-diagonal, consisting of $d/2$ independent 2D rotation blocks. We
 1071 can therefore analyze the contribution from a single block k and sum the results. Let $\theta_k = \omega_k^\top \Delta \mathbf{p}$.
 1072 The contribution from block k is:

$$1073 \quad \phi_k = (q_{2k-1} \quad q_{2k}) \begin{pmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

1076 Performing the matrix-vector multiplications, we get:

$$1077 \quad \phi_k = (q_{2k-1} \quad q_{2k}) \begin{pmatrix} \cos(\theta_k) - \sin(\theta_k) \\ \sin(\theta_k) + \cos(\theta_k) \end{pmatrix}$$

$$1078 \quad = q_{2k-1}(\cos(\theta_k) - \sin(\theta_k)) + q_{2k}(\sin(\theta_k) + \cos(\theta_k))$$

1080 Grouping terms by $\cos(\theta_k)$ and $\sin(\theta_k)$ reveals the linear projections:

$$\phi_k = \underbrace{(q_{2k-1} + q_{2k})}_{a_k(\mathbf{q}_i)} \cos(\theta_k) + \underbrace{(q_{2k} - q_{2k-1})}_{b_k(\mathbf{q}_i)} \sin(\theta_k)$$

1084 The Fourier coefficients $a_k(\mathbf{q}_i)$ and $b_k(\mathbf{q}_i)$ are thus simple linear combinations of the query vector's
1085 elements. Summing over all $k = 1, \dots, d/2$ yields the complete kernel $\phi_{\mathbf{q}_i}(\Delta\mathbf{p})$, which has the
1086 exact form stated in the proposition.

1088 C.2 PROOF OF COROLLARY 1

1089 The linear-time complexity is achieved by expressing the operation in matrix form and re-ordering
1090 the computation. Let $\mathbf{q}'_i = \rho_\Omega(\mathbf{p}_i)\mathbf{q}_i$ and $\mathbf{k}'_j = \rho_\Omega(\mathbf{p}_j)\mathbf{1}$. Let $\mathbf{Q}' \in \mathbb{R}^{N \times d'}$ be the matrix with rows
1091 $(\mathbf{q}'_i)^\top$, $\mathbf{K}' \in \mathbb{R}^{N \times d'}$ be the matrix with rows $(\mathbf{k}'_j)^\top$, and $\mathbf{V} \in \mathbb{R}^{N \times d_v}$ be the matrix of value vectors.
1092 The output matrix $\mathbf{Y} \in \mathbb{R}^{N \times d_v}$ is given by:

$$\mathbf{Y} = (\mathbf{Q}'(\mathbf{K}')^\top)\mathbf{V}.$$

1093 By the associativity of matrix multiplication, this can be computed as $\mathbf{Y} = \mathbf{Q}'((\mathbf{K}')^\top\mathbf{V})$. The term
1094 $(\mathbf{K}')^\top\mathbf{V}$ costs $O(Nd'd_v)$ to compute, resulting in a $d' \times d_v$ matrix. Multiplying this by \mathbf{Q}' costs an
1095 additional $O(Nd'd_v)$. The total complexity is therefore $O(Nd'd_v)$, linear in the sequence length N .
1096

1097 C.3 PROOF OF PLATONIC TRANSFORMERS IMPLEMENTING GROUP CONVOLUTIONS

1101 The dynamic convolution from Proposition 2 becomes a dynamic *group convolution* within the
1102 Platonic Transformer. This is a direct consequence of applying the operation to lifted coordinates
1103 $\mathbf{p}_i(R) = R^{-1}\mathbf{p}_i$ for each reference frame $R \in \mathcal{G}$. Since the relative position vector becomes
1104 $R^{-1}(\mathbf{p}_j - \mathbf{p}_i)$, the kernel's input is transformed accordingly. The resulting output for each frame
1105 takes the form of a group cross-correlation³:

$$\mathbf{y}_i(R) = \sum_{j=1}^N \phi_{\mathbf{q}_i(R)}(R^{-1}(\mathbf{p}_j - \mathbf{p}_i)) \mathbf{v}_j(R). \quad (21)$$

1106 Here, the kernel $\phi_{\mathbf{q}_i(R)}$ is steered by the group element R , defining an equivariant dynamic group
1107 convolution.

1112 D EQUIVALENT ATTENTION VIA ROPE BASE FREQUENCY STEERING

1114 We achieve full equivariance to Euclidean transformations by making the RoPE operator dependent
1115 on a local reference frame R by projecting positions \mathbf{p}_i on R to obtain positions $\mathbf{p}_i(R) := R^{-1}\mathbf{p}_i$.
1116 The attention scores $s_{ij}(R)$ for a query $\mathbf{q}(R)_i$ and key $\mathbf{k}(R)_j$ are computed as:

$$s_{ij}(R) = \mathbf{q}_i(R)^\top \rho_\Omega((\mathbf{p}_j - \mathbf{p}_i)(R)) \mathbf{k}_j(R), \quad (22)$$

1117 An equivalent approach is to steer the set of base RoPE frequencies Ω for each frame, creating a
1118 frame-specific set $\Omega_R = \{R\omega_k \mid \omega_k \in \Omega\}$ (Reddy & Chatterji, 1996). The attention scores are then
1119 computed as:

$$\hat{s}_{ij}(R) = \mathbf{q}_i(R)^\top \rho_{\Omega_R}(\mathbf{p}_j - \mathbf{p}_i) \mathbf{k}_j(R). \quad (23)$$

1123 *Proof.* For $s_{ij}(R)$ and $\hat{s}_{ij}(R)$ to be equivalent, we require that $\rho_\Omega((\mathbf{p}_j - \mathbf{p}_i)(R)) = \rho_{\Omega_R}(\mathbf{p}_j - \mathbf{p}_i)$.
1124 For this, we need to show that $\omega_k^\top \Delta\mathbf{p}(R) = (R\omega_k)^\top \Delta\mathbf{p}$, where $\Delta\mathbf{p} = \mathbf{p}_j - \mathbf{p}_i$. Let \mathbf{R} be the
1125 orthogonal matrix corresponding to R . Then we have:

$$\omega_k^\top \Delta\mathbf{p}(R) = \omega_k^\top R^{-1} \Delta\mathbf{p} \quad (24)$$

$$= \omega_k^\top \mathbf{R}^\top \Delta\mathbf{p} \quad (25)$$

$$= (\mathbf{R}\omega_k)^\top \Delta\mathbf{p} \quad (26)$$

$$= (R\omega_k)^\top \Delta\mathbf{p} \quad (27)$$

1131 Thus, projecting global positions or steering the base frequencies are equivalent. \square

1132
1133 ³Following common convention, we refer to this operation as a group convolution, though it is technically a
cross-correlation Cohen & Welling (2016); Bekkers (2020).

1134 By projecting the global positions, the RoPE attention mechanism remains identical to its traditional
 1135 formulation. Steering the base frequencies, however, is often more computationally efficient, since
 1136 the number of base frequencies is typically much smaller than the input sequence length.
 1137

1138 E FRAME AVERAGING

1140 Frame Averaging (Puny et al., 2022) (FA) imbues symmetry awareness to arbitrary neural networks.
 1141 A backbone model $\Phi : V \rightarrow W$ on normed spaces can be made equivariant (or invariant) to a group
 1142 \mathcal{G} . Specifically, for two group representations $\rho_1(g)$ and $\rho_2(g)$ that $g \in \mathcal{G}$ induces on V and W ,
 1143 $\Phi(\rho_1(g) \cdot \mathbf{x}) = \rho_2(g) \cdot \Phi(\mathbf{x})$, where $\mathbf{x} \in V$ and \cdot is the group action on elements from V and W .
 1144

1145 FA first constructs a *frame* $\mathcal{F}(\mathbf{x}) : V \rightarrow 2^{\mathcal{G}}$ with the following properties:

- 1146 • A frame is \mathcal{G} -equivariant if $\mathcal{F}(\rho_1(g)\mathbf{x}) = g\mathcal{F}(\mathbf{x})$, where $g\mathcal{F}(\mathbf{x}) = \{gh | h \in \mathcal{F}(\mathbf{x})\}$, and
- 1147 • A frame is bounded over a domain $K \subset V$ if $\exists c > 0$ such that the operator norm
 1148 $\|\rho_2(g)\|_{\text{op}} \leq c, \forall g \in \mathcal{F}(\mathbf{x}), \forall \mathbf{x} \in K$.

1150 To make the backbone Φ \mathcal{G} -equivariant, *symmetrization* is applied on \mathbf{x} via the group averaging
 1151 operator:

$$1153 \langle \Phi \rangle_{\mathcal{F}}(\mathbf{x}) = \frac{1}{|\mathcal{F}(\mathbf{x})|} \sum_{g \in \mathcal{F}(\mathbf{x})} \rho_2(g)\Phi(\rho_1(g)^{-1}(\mathbf{x})). \quad (28)$$

1155 For equivariance to 3-dimensional Euclidean rigid-body transformations (translations, rotations,
 1156 reflections), i.e., $\mathcal{G} := E(3)$ and $V = \mathbb{R}^3$, $\mathcal{F}(\mathbf{x})$ is constructed using PCA on \mathbf{x} ; this involves
 1157 computing the centroid $\mathbf{t} = \frac{1}{n}\mathbf{x}^\top \mathbf{1} \in \mathbb{R}^3$ and covariance matrix $\mathbf{C} = (\mathbf{x} - \mathbf{1}\mathbf{t}^\top)^\top(\mathbf{x} - \mathbf{1}\mathbf{t}^\top)$ for a
 1158 point cloud $\mathbf{x} \in \mathbb{R}^{n \times 3}$ with n points. Suppose we obtain eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in \mathbb{R}^3$ of \mathbf{C} , we
 1159 construct 3×3 orthogonal matrices by concatenating them together. Depending on the collection of
 1160 these matrices, we achieve equivariance to different motion groups: for $E(3)$ equivariance, we use
 1161 $\mathbf{U} = [\pm \mathbf{u}_1, \pm \mathbf{u}_2, \pm \mathbf{u}_3] \subset E(3)$, and if we restrict this collection to contain only orthogonal, positive
 1162 orientation matrices, we achieve $SE(3)$ equivariance. The frame then looks like,
 1163

$$1164 \mathcal{F}(\mathbf{x}) = \{(\mathbf{U}, \mathbf{t}) : \mathbf{U} = [\pm \mathbf{u}_1, \pm \mathbf{u}_2, \pm \mathbf{u}_3]\} \subset E(3).$$

1165 For equivariant predictions on atomistic point clouds, we set $\rho_1(g)\mathbf{x} = \mathbf{x}\mathbf{U}^\top + \mathbf{1}\mathbf{t}^\top$ and $\rho_2(g)\mathbf{x} =$
 1166 $\mathbf{x}\mathbf{U}^\top$ for each frame element, followed by the application of the averaging operator in Eq. 28.
 1167 Furthermore, Puny et al. (2022) show that if $\mathcal{F}(\mathbf{x})$ is bounded, FA preserves the expressiveness of the
 1168 underlying backbone, making $\langle \Phi \rangle_{\mathcal{F}}$ maximally expressive even for non-compact groups like $E(n)$.
 1169

1170 F DETAILS OF ARCHITECTURE

1171 In this section, we provide additional details about the architecture of the Platonic Transformer
 1172 and the various model configurations used in our experiments. Our framework is designed to be
 1173 equivariant to roto-translation groups, primarily $SE(n)$ and, through specific configurations, the full
 1174 Euclidean group $E(n)$.
 1175

1176 We denote the core embedding dimension per group element as d_{hidden} . Since our features are
 1177 functions on a group \mathcal{G} of order $|\mathcal{G}|$, the total feature dimension of a layer is $d_{\text{model}} = |\mathcal{G}| \times d_{\text{hidden}}$.
 1178 The specific group is determined by the `solid_name` parameter.
 1179

1180 For the initial feature processing, input scalars and vectors are first embedded and then lifted into
 1181 a group-equivariant feature space using an initial lifting operation. This creates a tensor where the
 1182 channel dimension is expanded by a factor of $|\mathcal{G}|$. An initial group-equivariant linear layer then
 1183 projects these lifted features to the model's working dimension, d_{model} . Optionally, an equivariant
 1184 Absolute Positional Encoding (APE), parameterized by `ape_sigma`, can be added at this stage.
 1185

1186 The main body of the network consists of a stack of equivariant transformer blocks. Each block
 1187 contains two main sub-modules: a group-equivariant interaction layer and a feed-forward network
 1188 (FFN), connected with residual connections. Normalization is applied either before each sub-module
 1189 or after.

1188 For the group-equivariant interaction layer, we denote the number of attention heads per group
 1189 element as n_{head} . The total number of effective parallel heads is therefore $|\mathcal{G}| \times n_{\text{head}}$. The dimension
 1190 of each head, d_{head} , is calculated as $d_{\text{hidden}}/n_{\text{head}}$. The input features are first projected to query,
 1191 key, and value representations using group-equivariant linear layers. To encode relative spatial
 1192 information, group-equivariant Rotary Position Embeddings, parameterized by `rope_sigma` and
 1193 `learned_freqs`, are applied to the query and key vectors. The interaction can then be performed
 1194 either as a full softmax-based attention mechanism or as a linear-time dynamic group convolution
 1195 via the `attention` flag. For the Feed Forward Networks (FFNs), we denote the hidden feature
 1196 dimension as $d_{\text{ffn}} = d_{\text{model}} \times f_{\text{factor}}$. The FFN consists of two group-equivariant linear layers with a
 1197 GELU activation function in between.

1198 For the final output, two separate readout heads project the features to the desired scalar and vector
 1199 output dimensions. For graph-level tasks, a pooling operation performs a mean aggregation over the
 1200 node and group dimensions to produce a final invariant prediction. For node-level tasks, an averaging
 1201 operation over the group axis projects the features back to standard invariant scalar and equivariant
 1202 vector representations. Following standard Transformer practices, we apply dropout to the attention
 1203 weights and FFN activations, and stochastic depth to the outputs of the equivariant transformer blocks.

1204 Particular values for all the important hyperparameters used for the experiments are in the Table.8
 1205

1206 G HYPERPARAMETER TUNING AND MODEL SELECTION STRATEGY

1208 This section outlines the full procedure used to configure and train our models.

1209
 1210 **Baseline Optimization** To establish a fair point of comparison, we first optimized the general
 1211 training protocol using only the translation-only equivariant ($T(n)$) models. This initial phase
 1212 involved tuning the optimizer, learning rate schedule, weight decay, and data augmentations to ensure
 1213 the baseline models were as competitive as possible. This fixed protocol was then used for all
 1214 subsequent experiments.

1215
 1216 **Hyperparameter Sweep for Model Selection** With the training protocol fixed, we performed an
 1217 extensive hyperparameter sweep for both $SE(n)$ and $T(n)$ model classes. This sweep was designed
 1218 to find the optimal architectural parameters while maintaining an equal computational budget between
 1219 model families. The parameters and their swept values are summarized in Table 6.

1220
 1221 Table 6: Hyperparameter Sweep Configurations.

1222 1223 Parameter	1224 Configuration	1225 Values
1224 Hidden Dim	-	[384, 576, 768, 1152]
1225 Number of Heads	$T(n)$ model (HS=16) $T(n)$ model (HS=32) $SE(n)$ model (HS=16) $SE(n)$ model (HS=32)	[24, 36, 48, 72] [12, 18, 24, 36] [2, 3, 4, 6] heads per group element [1, 2, 3] heads per group element
1226 Rope Sigma (σ_{rope})	RoPE frequency scaling	[0.5 – 2.0]
1227 Attention	Full Attention / Linear Conv	[True, False]
1228 Solid Group (\mathcal{G})	Symmetry group	[Octahedron, Tetrahedron, C_{2-8} , D_{4-8}]
1229 Lambda F (λ_F)	OMol Force loss weight	[1.0 – 25.0]
1230 Batch Size	Samples per batch	[64 – 512]
1231 Weight Decay		[$1e^{-3}$ – $1e^{-7}$]

1232
 1233 The hyperparameter sweep was conducted across multiple layers and three random seeds for a
 1234 *moderate number of epochs* to efficiently explore the configuration space. It should be noted that
 1235 some configurations are not applicable for the Octahedral group, as its 24 symmetry elements require
 1236 a minimum of 24 total effective heads (i.e., at least one head per group element). After identifying the

best-performing hyperparameters for both the $SE(n)$ and $T(n)$ model families from this sweep, we proceeded to a final, full-length training run. These selected models were trained for a *large number of epochs* to ensure convergence again with fixed compute budget, producing the final results reported in the main paper.

H DETAILS OF EXPERIMENTS ON CIFAR10

H.1 DESCRIPTION OF THE DATASET

The CIFAR-10 dataset (Krizhevsky, 2009) is a standard benchmark for image classification, consisting of 60,000 32x32 color images across 10 classes. The dataset is divided into a training set of 50,000 images and a test set of 10,000 images.

H.2 TRAINING DETAILS

For the CIFAR-10 classification task, our experimental setup is closely adapted from the supervised training recipe for Vision Transformers presented in DeiT-III (Touvron et al., 2022). We tokenize each image into a sequence of non-overlapping patches using a patch size of 4×4 pixels, a key deviation from the ImageNet configurations to suit the lower resolution of the dataset.

The model is trained using the LAMB optimizer, which is subject to a cosine decay schedule following a 5-epoch warm-up period. A comprehensive suite of regularization techniques is employed, including a weight decay of 0.02, Mixup with an alpha value of 0.8, and CutMix with an alpha of 1.0, in addition to model-size-dependent Stochastic Depth. The data augmentation pipeline is built upon the ‘3-Augment’ strategy, incorporating standard Random Resized Crop (RRC), horizontal flips, ColorJitter with a factor of 0.3, and a single, randomly selected transformation from a pool of three: Grayscale, Solarization, or Gaussian Blur.

The training objective is optimized using a Binary Cross-Entropy (BCE) loss, and positional information is supplied to the transformer blocks through a combination of both Absolute Positional Encodings (APE) and Rotary Position Embeddings (RoPE). Further hyperparameter details are available in Table 8.

I DETAILS OF EXPERIMENTS ON SCANOBJECTNN

I.1 DESCRIPTION OF THE DATASET

ScanObjectNN(Uy et al., 2019) dataset is a real-world 3D point cloud dataset. It contains 15,000 objects divided into 15 categories with 2902 unique object instances. It contains background, parts missing, and object deformation elements, which makes the classification task a challenge. The dataset consists of three variants OBJ_BG, OBJ_ONLY and PB_T50_RS, for now the latter is only examined.

I.2 TRAINING DETAILS

In order to prepare the input point cloud $\mathbf{P} \in \mathbb{R}^{N \times 3}$ for processing by the Platonic Transformer, we follow a preprocessing procedure. Similar to established methods Pang et al. (2023); Yu et al. (2022), we first use Farthest Point Sampling (FPS) to select a set of $L=2048$ central points, denoted as $\mathbf{P}_C \in \mathbb{R}^{L \times 3}$ with $L = 2048$. Subsequently, for each central point P_C^i , we define a local patch $x_p^i \in \mathbb{R}^{K \times 3}$ by identifying its K-Nearest Neighbors (KNN) within the original point cloud \mathbf{P} . These local patches serve as the primary input vectors to the Platonic Transformer.

Additionally, to account for the axis-aligned nature of the dataset and to provide the model with a global reference frame, we incorporate rotation augmentation. For each input vector, a rotation matrix is applied. This matrix is either a random rotation or the 3×3 identity matrix, which is concatenated with the input vector to provide the model with information about the global orientation.

1296 **J DETAILS OF EXPERIMENTS ON QM9**
12971298 **J.1 DESCRIPTION OF THE DATASET**
12991300 The QM9 dataset (Ramakrishnan et al., 2014) contains up to 9 heavy atoms and 29 atoms, including
1301 hydrogens. We use the train/val/test partitions introduced in Gilmer et al. (2017), which consist of
1302 100K/18K/13K samples, respectively, for each partition.
13031304 **J.2 TRAINING DETAILS FOR THE REGRESSION EXPERIMENT**
13051306 For the QM9 regression task, we train the Platonic Transformer to predict molecular properties.
1307 Before being fed to the model, the input molecular geometries are centered by subtracting the mean
1308 coordinate of each molecule. To stabilize training, we normalize the target property values by
1309 subtracting their mean and dividing by their standard deviation, with these statistics computed over
1310 the training set. We employ data augmentation in the form of random $SO(3)$ rotations applied to the
1311 coordinates during training.
13121313 The model is trained for a total of 1000 epochs using a batch size of 96. We utilize the Adam
1314 optimizer with a learning rate of 5×10^{-4} and a weight decay of 10^{-8} . A cosine annealing schedule
1315 with a 10-epoch linear warmup adjusts the learning rate throughout training. To prevent exploding
1316 gradients, we apply gradient clipping with a maximum norm of 0.5. The training objective is
1317 the Mean Absolute Error (MAE) on the normalized target values, while validation and testing are
1318 performed by calculating the MAE on the original, unnormalized scale. Our experiments explore
1319 different Platonic Transformer configurations, specifically by varying the symmetry group among
1320 `trivial_3`, `tetrahedron`, and `octahedron`. Further hyperparameter details are available in
1321 Table 8.
13221323 We choose FAFormer (Huang et al., 2024) to compare the contribution of Frame Averaging (with
1324 a standard Transformer backbone) with our end-to-end trained Platonic Transformer with different
1325 platonic solid symmetry groups. Recent efforts in long-context sequence modeling have also pit
1326 Transformer-based methods with state-space methods. To provide a similar analysis here, we choose
1327 G-Hyena (Moskalev et al., 2025), a recent state-space model that relies on long convolutions for 3D
1328 molecular point clouds. We hyperparameter-tune FAFormer⁴ and G-Hyena to have similar parameter
1329 counts and representational capacity as our best-performing Platonic Transformer on QM9. The
1330 baselines are also trained over 500 epochs with a batch size of 96 on a single H200 GPU.
13311332 **J.3 ADDITIONAL DETAILS ON WALL-CLOCK TIMINGS**
13331334 To produce the wall-clock timing for the standard Transformer in Table 4, node features from $B = 64$
1335 QM9 molecules were projected from $d_{\text{in}} = 11$ to $d_{\text{model}} = 512$ using a Linear layer and fed as tokens
1336 into a `TransformerEncoderLayer` module provided by PyTorch with 16 heads. We measure
1337 wall-clock timings for a forward pass over 10 batches on a single H200 GPU. This offers a reference
1338 timing to compare the inference speed of the Platonic Transformer and other geometric baselines like
1339 G-Hyena (Moskalev et al., 2025) and the Tensor Field Network (Thomas et al., 2018).
13401341 **J.4 EXTENDED EXPERIMENTS ON QM9**
13421343 We extend our evaluation on the QM9 dataset to include all property targets. Table 3 summarizes these
1344 results. We observe that the relative performance hierarchy remains consistent with our main findings:
1345 models with higher-order symmetry groups (Octahedron) generally outperform those with lower
1346 symmetry (Tetrahedron) and the non-equivariant baseline (Trivial). This confirms that the benefits of
1347 the Platonic Transformer’s geometric inductive bias generalize across different properties. Crucially,
1348 unlike baselines such as EquiformerV2 which rely on target-specific hyperparameter tuning, we
1349 employ a single fixed set of hyperparameters across all targets. Despite this constraint, the Platonic
1350 Transformer achieves competitive results, suggesting that further performance gains could be realized
1351 with target-specific optimization.
13521353 ⁴We use the open-source implementation of FAFormer provided at <https://github.com/Graph-and-Geometric-Learning/Frame-Averaging-Transformer>. We obtain the source
1354 code for G-Hyena through private correspondence with the authors.
1355

1350 Table 7: Mean absolute error results on QM9 test set. \dagger denotes using different data partitions.
 1351 Missing entries will be completed before the camera ready.

Model	Task Units	α_{3}	$\Delta\epsilon_{\text{meV}}$	$\epsilon_{\text{HOMO meV}}$	$\epsilon_{\text{LUMO meV}}$	μ_{D}	C_{ν} cal/mol K	G_{meV}	H_{meV}	$R^2_{a_0^2}$	U_{meV}	U_0_{meV}	ZPVE meV
DimeNet++ (Gasteiger et al., 2020)	.044	33	25	20	.030	.023	8	7	.331	6	6	1.21	
EGNN (Satorras et al., 2021) [†]	.071	48	29	25	.029	.031	12	12	.106	12	11	1.55	
PaiNN (Schütt et al., 2021)	.045	46	28	20	.012	.024	7.35	5.98	.066	5.83	5.85	1.28	
TorchMD-NET (Thölke & Fabritiis, 2022)	.059	36	20	18	.011	.026	7.62	6.16	.033	6.38	6.15	1.84	
SphereNet (Liu et al., 2022)	.046	32	23	18	.026	.021	8	6	.292	7	6	1.12	
SEGNN (Brandstetter et al., 2022) [†]	.060	42	24	21	.023	.031	15	16	.660	13	15	1.62	
EGOAT (Le et al., 2022)	.053	32	20	16	.011	.024	23	24	.382	25	25	2.00	
Equiformer (Liao & Smidt, 2023)	.046	30	15	14	.011	.023	7.63	6.63	.251	6.74	6.59	1.26	
EquiformerV2 (Liao et al., 2024)	.050	29	14	13	.010	.023	7.57	6.22	.186	6.49	6.17	1.47	
PÖNITA (Bekkers et al., 2024)	.038	30.4	16.0	14.5	.012	.024	8.63	8.04	.235	8.67	8.31	1.29	
Platonic Transformer (Trivial, Attn)	.064	$45.9_{\pm 0.18}$	$29.4_{\pm 0.50}$	$24.4_{\pm 0.35}$.028	-	-	-	-	-	-	-	
Platonic Transformer (Trivial, Conv)	.061	$43.8_{\pm 0.65}$	$26.6_{\pm 0.09}$	$24.0_{\pm 0.29}$.030	$.033_{\pm .0006}$	-	-	$.256_{\pm .0044}$	-	-	-	
Platonic Transformer (Tetra, Attn)	.049	-	-	-	.012	-	-	-	-	-	-	-	
Platonic Transformer (Tetra, Conv)	.047	-	-	-	.014	-	-	-	-	-	-	-	
Platonic Transformer (Octa, Attn)	$.049_{\pm .0007}$	$37.4_{\pm 1.36}$	$22.2_{\pm 1.21}$	$16.7_{\pm 0.42}$	$.010_{\pm .0002}$	$.024_{\pm .0001}$	$12.0_{\pm 1.00}$	$12.0_{\pm 0.26}$	$.222_{\pm .0062}$	$11.9_{\pm 1.73}$	$13.0_{\pm 0.09}$	$1.3_{\pm 0.01}$	
Platonic Transformer (Octa, Conv)	$.048_{\pm .0013}$	$33.8_{\pm 1.00}$	$17.7_{\pm 0.51}$	$15.7_{\pm 0.25}$	$.012_{\pm .0001}$	$.026_{\pm .0001}$	$11.0_{\pm 0.45}$	$11.7_{\pm 0.20}$	$.184_{\pm .0110}$	$13.9_{\pm 0.85}$	$10.9_{\pm 0.36}$	$1.4_{\pm 0.03}$	

Table 8: Hyperparameter for all datasets

Hyperparameter	QM9	OMol25	CIFAR10	ScanNetNN
Architecture				
hidden_dim	1152	1152	768	576
layers	14	14	12	12
num_heads	72	72	12	12
Positional encoding				
freq_sigma	0.5	0.5	1	18
ape_sigma	0.5	None	10	10
learned_freqs	True	True	True	True
Attention / readout				
dropout	0.0	0.0	0.0	0.1
drop_path_rate	0.0	0.0	0.1	0.0
mean_aggregation	False	False	False	False
Training				
lr	5e-4	2e-4	8e-4	8e-4
batch_size	96	64	256	128
epochs	1000	22	500	500
warmup	10	5	20	10
weight_decay	1e-8	1e-6	0.05	1e-6
lambda_F	-	12.0	-	-
cosine_scheduler	True	True	True	True
precision	32	32	32	32
gpus	1	4	1	1

K DETAILS OF EXPERIMENTS ON OMOL25

K.1 DESCRIPTION OF THE DATASET

For large-scale molecular experiments, we use the Open Molecules 2025 (OMol25) dataset (Levine et al., 2025), a comprehensive collection of over 100 million Density Functional Theory (DFT) calculations performed at the wB97M-V/def2-TZVPD level of theory. This dataset is notable for its vast chemical and structural diversity, encompassing 83 elements and systems up to 350 atoms. The structures are drawn from a wide range of chemical domains, including small molecules, biomolecules, metal complexes, and electrolytes, and feature varied charges, spin states, conformers, and reactive geometries.

1404 The OMol25 dataset is organized into several training sets and splits for validation and testing
 1405 to ensure consistent and robust model evaluation. The full training set, "All," contains over 100
 1406 million DFT calculations. For more computationally efficient training and development, a smaller,
 1407 uniformly sampled "4M" split is provided, containing approximately 4 million structures. Our work
 1408 primarily utilizes the "Neutral" split, which consists of approximately 34 million charge-neutral,
 1409 singlet structures drawn from established community datasets like ANI-2X, GEOM, and SPICE2.
 1410 This split is designed to benchmark model performance on familiar organic chemistry space without
 1411 the added complexity of variable charge and spin.

1412 For validation and testing, OMol25 provides several out-of-distribution (OOD) splits designed to
 1413 evaluate model generalizability. The primary validation set ("Val Comp") consists of structures with
 1414 compositions held out from the training set. Further specialized test sets include held-out organic and
 1415 metal-complex reactions ("Test Reactivity"), experimental crystal structures from the Crystallography
 1416 Open Database ("Test COD"), and unique anion structures ("Test Anions"), among others. The core
 1417 task is Structure to Energy and Forces (S2EF), where models are evaluated on their ability to predict
 1418 the total energy of a structure and the per-atom forces, with Mean Absolute Error (MAE) being the
 1419 primary metric.

1420 K.2 TRAINING DETAILS

1421 We partition the "Neutral" split of the OMol25 dataset into an 80% training set and a 20% validation
 1422 set, and report the final results on the official test set. The model is trained using an AdamW optimizer
 1423 with a learning rate of 2×10^{-4} and a weight decay of 10^{-6} . The learning rate is managed by a
 1424 cosine decay schedule, which includes a linear warmup period over the first 5 epochs. Training is
 1425 conducted for a total of 22 epochs using a batch size of 64.

1426 The training objective is a weighted sum of two components: a **Mean Squared Error (MSE)** loss
 1427 for the total energy and a **Mean Absolute Error (MAE) on the force vectors**. The force loss is
 1428 calculated as the average L2 norm (Euclidean distance) of the error between the predicted and target
 1429 force vectors for each atom. To balance these two targets, the force loss component is weighted by a
 1430 factor of $\lambda_F = 12.0$.

1431 To ensure stable training on this large-scale task, we normalize the target energies. We apply a linear
 1432 referencing scheme to the raw DFT energies. This method normalizes the total energy by subtracting
 1433 the pre-computed DFT energies of the constituent isolated atoms:

$$1436 E_{\text{ref}} = E_{\text{DFT}} - \sum_{i=1}^N E_i^{\text{atom}} \quad (29)$$

1437 where E_{ref} is the target value for the model, E_{DFT} is the system's total energy, N is the number of
 1438 atoms, and E_i^{atom} is the pre-computed DFT energy of an isolated atom of the same species as atom i .
 1439 This procedure is consistent with the methodology used for the OC22 dataset (Tran et al., 2023) and
 1440 helps maintain comparability with other large-scale models.

1441 Given the significant computational requirements for training on OMol25, we established a fixed
 1442 compute budget to ensure a fair comparison between models. Each model was trained for a max-
 1443 imum duration of 5 days on a node equipped with 4x NVIDIA 6000Ada GPUs. The detailed
 1444 hyperparameters for our model configuration on this dataset are summarized in Table 8.

1445

1446 L EXPERIMENTS ON LEARNED KEY PROJECTIONS

1447 In Section 4.1 and Remark 1 of the main text, we motivate our design choice of using fixed key vectors
 1448 ($k_j = 1$)—as opposed to learned linear projections ($k_j = W^K f_j$)—by citing training instabilities
 1449 observed on molecular datasets. In this section, we report an empirical analysis of this observation.

1450

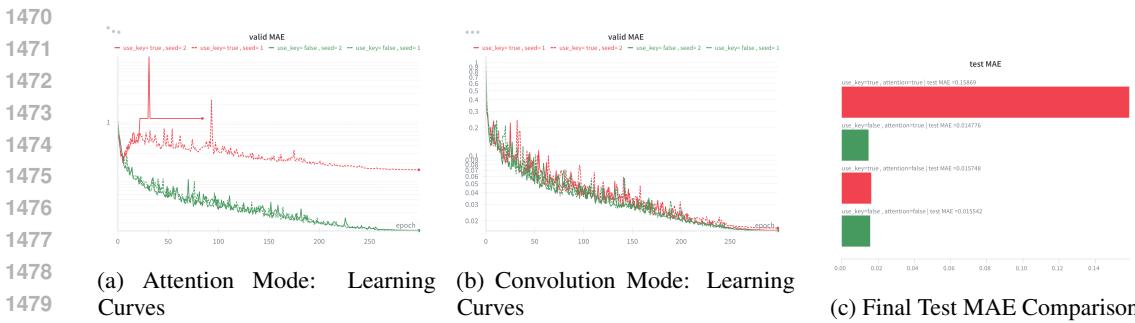
1451 L.1 INSTABILITY OF LEARNED KEYS

1452 To investigate the impact of learned keys, we conducted experiments on the QM9 dataset using
 1453 the standard hyperparameters defined in Appendix J. We compared the standard model (fixed keys)
 1454 against a variant with learned key projections. We performed this comparison for both the full

1458 Attention mechanism and the linear Convolutional variant, training for 300 epochs across two random
 1459 seeds.

1460 The results are illustrated in Figure 4. As shown in Figure 4a, when using the full Attention
 1461 mechanism, the introduction of learned keys ('use_key=True') leads to severe training instability.
 1462 Both runs utilizing learned keys exhibit divergence around epoch 10, with one run failing to complete.
 1463 In contrast, the fixed key formulation ('use_key=False') trains smoothly.

1464 In the linear Convolutional mode (Figure 4b), training remains stable for both configurations. How-
 1465 ever, as shown in Figure 4c, the learned keys provide no performance benefit; in fact, the model with
 1466 fixed keys achieves a lower Test MAE. This suggests that even when stability is maintained, the
 1467 entanglement of content and geometry introduced by learned keys does not improve generalization
 1468 for this physical task.

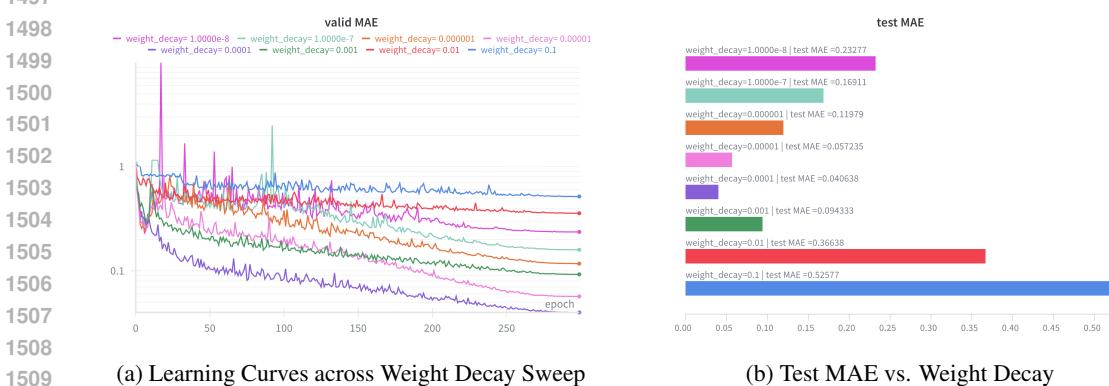


1481 Figure 4: Impact of Learned Key Projections on Stability and Performance. (a) When using full
 1482 attention, learned keys cause rapid divergence/instability around epoch 10. (b) In convolutional mode,
 1483 training is stable, but (c) fixed keys consistently outperform learned keys in final accuracy.

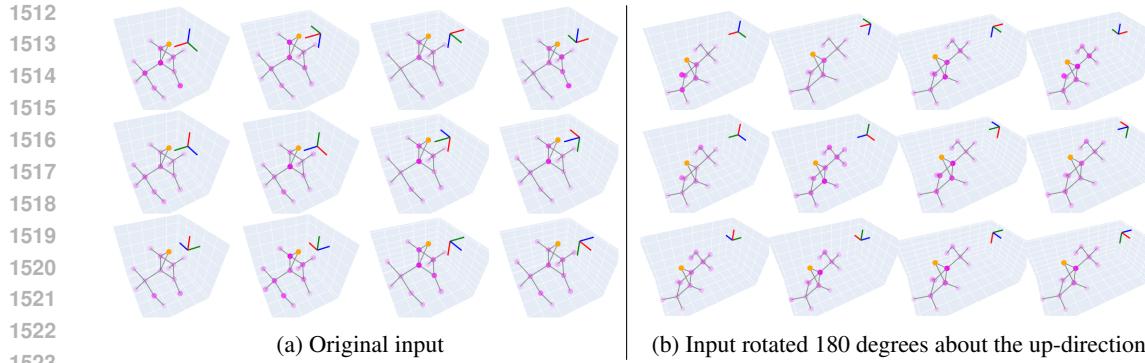
1486 L.2 MITIGATING INSTABILITY VIA REGULARIZATION

1487 We further hypothesized that the instability in the Attention setting might be mitigated by stronger
 1488 regularization. We performed a sweep of weight decay values ranging from 10^{-1} to 10^{-8} for the
 1489 model with learned keys.

1490 Figure 5 presents these results. Figure 5a shows that while high weight decay values (10^{-1} to
 1491 10^{-4}) can stabilize the training, reducing the weight decay below 10^{-4} immediately reintroduces the
 1492 instability observed in the previous experiment. Figure 5b shows that the the weight decay should
 1493 be as small as possible while still leading to stable training, however, still the best performance is
 1494 lagging considerably behind the results of our default weight decay setting of 10^{-8} for the constant
 1495 key scenario (Figure 4).



1509 Figure 5: Can Weight Decay Fix Learned Keys? (a) Strong weight decay stabilizes training, while
 1510 values $< 10^{-4}$ lead to divergence. (b) The final test MAEs of each model.



1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524 We visualize the attention score between the orange node and all others, where an increased
1525 color intensity indicates an increased attention score. The subplots correspond to 12 different
1526 frames in the same head of an octahedral Platonic Transformer layer (there are 12 more frames not
1527 visualized here). The attention is broadly focused on locality but with distinct directional biases. The
1528 equivariance of the model can be observed by comparing the attention scores in the sub-figures. For
1529 instance, the attention pattern in the top-left frame in Figure 6a is the same as the one in the top-right
1530 frame in Figure 6b, but rotated 180 degrees.

1531
1532 **Conclusion:** These experiments confirm that for physical tasks like QM9, using fixed keys ($k = 1$)
1533 is not merely a simplification but an important design choice that ensures training stability and
1534 solid performance. While we report results over 300 epochs here (compared to 1000 in the main
1535 experiments), the early onset of instability and the consistent performance gap makes it implausible
1536 that heavy regularized learned-key platonic transformers could match the constant-key performance.
1537

1539 M VISUALIZATIONS OF LEARNED ATTENTION SCORES

1541 To show the directional attention learned in the attention head, we visualize examples over attention
1542 patterns in different frames $g \in \mathcal{G}$ in Figure 6.

1544 N IMPLEMENTING PLATONIC TRANSFORMERS IN THE FOURIER DOMAIN OF 1545 FINITE GROUPS

1548 With increasing hidden dimension (while not increasing sequence length), transformer blocks spend
1549 more and more of their total compute time in the pointwise linear layers. To improve speed it can
1550 then be worthwhile to implement the pointwise equivariant linear layers in the Fourier domain of
1551 the rotation group, a technique that has recently been successfully employed in computer vision
(Bökman et al., 2025; Nordström et al., 2025). Considering the Fourier domain also sheds light on the
1552 connections between Platonic Transformers and equivariant networks with general steerable feature
1553 spaces (Cesa et al., 2022).

1555 In this section we demonstrate how a Fourier domain implementation can improve computational
1556 efficiency in Platonic Transformers. In the Fourier domain, equivariant linear layers are block-
1557 diagonal, drastically reducing the required number of FLOPs for both forward and backward passes.
1558 We will see that with the number of hidden dimensions considered in this paper, a naive PyTorch
1559 implementation is not efficient enough to realize the reduction in FLOPs in terms of a substantial
1560 reduction in training throughput, but at a moderately higher number of hidden dimensions, there are
1561 throughput gains. This suggests that future scaling of Platonic Transformers will benefit from being
1562 implemented in the Fourier domain, and that more efficient implementations than our current one
1563 would be able to improve throughput even at smaller number of hidden dimensions.

1564 We will use the tetrahedral symmetry group as a running example in this section. The reader is
1565 cautioned that the representations discussed in this section are representations of the rotation group,
in contrast to the representations of the translation group discussed in Appendix A.

1566
1567

N.1 INTRODUCTION TO THE FOURIER THEORY OF FINITE GROUPS

1568
1569
1570
1571

The representation theory of finite groups is a well studied topic with many good text books. We recommend (Serre, 1977) for more detailed background than given here. Note that we consider vector spaces over the real numbers, which leads to a slightly more involved representation theory than complex numbers, see (Serre, 1977, Section II.12).

1572
1573
1574
1575
1576
1577
1578
1579

Recall from Appendix A.1 that a representation of a group \mathcal{G} is a group homomorphism $\rho : \mathcal{G} \rightarrow GL(V)$, where V is a vector space. We will here consider finite real vector spaces $V = \mathbb{R}^n$ so that $\rho(g)$ can be considered real-valued invertible matrices. An irreducible representation is one where the matrices $\{\rho(g)\}_{g \in \mathcal{G}}$ can not be simultaneously block-diagonalized. Any finite group \mathcal{G} has a finite number (up to isomorphisms) of irreducible representations (irreps) $\{\rho_i\}$ and they can be computed given the multiplication table of the group. Irreps are important because we can decompose any finite representation ρ into a direct sum of irreps by performing a change of basis, so statements about general representations often reduce to statements about irreps.

1580
1581
1582
1583

The features in Platonic Transformers are functions from \mathcal{G} to \mathbb{R}^C , that transform under the left regular representation as explained in Appendix B. In other words, the representation that acts on them is a direct sum of C copies of the regular representation of \mathcal{G} . Let this representation be denoted $\tilde{\rho}$. Decomposing $\tilde{\rho}$ into irreps, we obtain

1584
1585
1586

$$\tilde{\rho}(g) = Q \left(\bigoplus_i \rho_i(g)^{\oplus m_i} \right) Q^{-1} \quad (30)$$

1587
1588

for some multiplicities m_i of each irrep and a change of basis matrix Q that can be taken to be orthogonal.

1589
1590
1591
1592
1593
1594
1595

Now, Schur's lemma says that any equivariant linear map between non-isomorphic irreps $\rho_i \neq \rho_j$ must be constant zero. Further, the space of equivariant linear maps between ρ_i and itself is 1-, 2-, or 4-dimensional and isomorphic (as a division algebra over \mathbb{R}) to the real numbers, complex numbers, or quaternions depending on whether ρ_i is of so-called real, complex or quaternion type. (The type of ρ_i can be computed.) This means that any linear map that is equivariant from $\tilde{\rho}$ to $\tilde{\rho}$ is actually block diagonal after having performed the change of basis in (30), in particular so are the group convolutions used in Platonic Transformers.

1596
1597
1598

For cyclic groups, the block-diagonalization corresponds to the fact that convolutions are pointwise multiplications in the Fourier domain⁵.

1599
1600

N.2 FOURIER THEORY OF THE TETRAHEDRAL GROUP

1601
1602
1603

Let us now consider the Tetrahedral rotation group as \mathcal{G} , consisting of the twelve rotational symmetries of a regular tetrahedron. This group is isomorphic to the alternating group A_4 and has three real irreps. The real irreps of the tetrahedral group are given by the one-dimensional trivial representation

1604
1605
1606

$$\rho_1(R) = 1, \quad (31)$$

the three-dimensional standard representation

1607

$$\rho_3(R) = R \quad (32)$$

1608
1609
1610

and a two-dimensional representation ρ_2 that is defined as follows. Note that any element in \mathcal{G} is either the identity, a rotation by $2\pi/3$ radians (there are 8 of these) or a rotation by π radians (there are 3 of these). For the identity and rotations by π ,

1611
1612
1613

$$\rho_2(R) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (33)$$

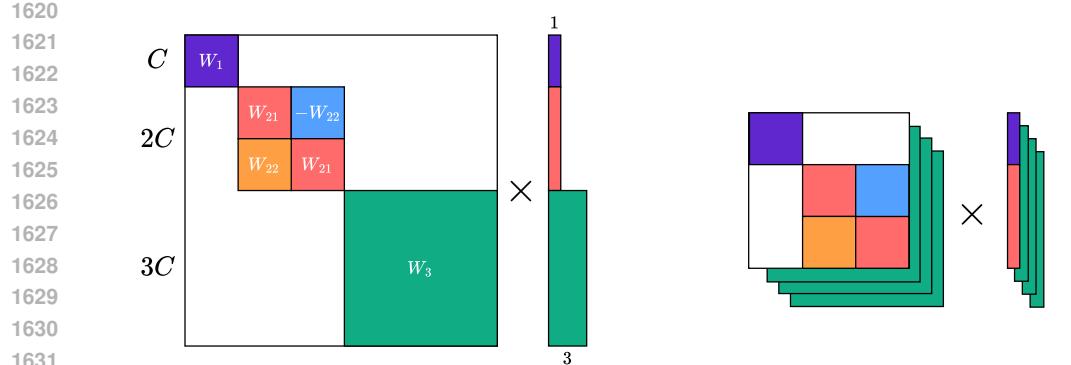
1614
1615
1616
1617
1618

The rotations by $2\pi/3$ fall into two conjugacy classes of four elements each, where one conjugacy class contains the inverses of the second. We can arbitrarily choose one of the conjugacy classes and define

$$\rho_2(R) = \begin{pmatrix} \cos(2\pi/3) & -\sin(2\pi/3) \\ \sin(2\pi/3) & \cos(2\pi/3) \end{pmatrix} \quad (34)$$

1619

⁵This requires working over the complex numbers, over the real numbers the pointwise multiplications turn into 2×2 matrix multiplications, again a block-diagonal structure.



1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
(a) The block-diagonal structure of an equivariant weight matrix in the Fourier domain.

(b) We can implement the linear layer as a batched matrix-vector multiplication with four batches.

Figure 7: We visualize the weight matrices for linear layers that are equivariant under the tetrahedral rotation group, implemented in the Fourier domain. Each subfigure shows weights to the left and features to the right. Purple features transform according to ρ_1 (or technically $\rho_1 \otimes I_C$ since there are C copies of ρ_1), red features according to ρ_2 (by multiplication by $\rho_2(g) \otimes I_C$ from the left) and green features according to ρ_3 (by multiplication by $\rho_3(g)^\top$ from the right (if we flattened the green features, they would transform by $\rho_3(g) \otimes I_{3C}$ from the left)). The weight matrix is parameterized by the $C \times C$ matrices W_1 , W_{21} , W_{22} and the $3C \times 3C$ matrix W_3 , yielding a total of $12C^2$ learnable parameters. The total number of multiplications to compute the linear layer implemented as a batched matrix-multiplication in 7b is $4 \cdot (3C)^2 = 36C^2$, yielding a $4 \times$ FLOP reduction versus an ordinary layer from $12C$ to $12C$ dimensions ($144C^2$ multiplications).

there, which implicitly defines the values for the second conjugacy class to be the inverse of the above.

It can be computed that ρ_1 and ρ_3 are both of real type, while ρ_2 is of complex type. Hence, equivariant linear maps from ρ_1 to ρ_1 are parameterized by one value, and the same for ρ_3 . Equivariant linear maps from ρ_2 to ρ_2 are instead parameterized by two values (this is because ρ_2 splits into two irreps over the complex numbers).

It can also be computed (or recovered from general facts of the Fourier transform over finite groups) that the representation $\tilde{\rho}$ acting on features with C channels in a tetrahedral Platonic Transformer splits into C copies of ρ_1 , C copies of ρ_2 and $3C$ copies of ρ_3 (as a sanity check, we recover all $C + C \cdot 2 + 3C \cdot 3 = 12C$ dimensions).

As mentioned, Schur's lemma now implies that equivariant linear maps from $\tilde{\rho}$ to itself are block-diagonal. The map from copies of ρ_1 to copies of ρ_1 is parameterized by a $C \times C$ matrix, the map from copies of ρ_2 to copies of ρ_2 is parameterized by two $C \times C$ matrices (because ρ_2 is of complex type) and the map from copies of ρ_3 to copies of ρ_3 is parameterized by a $3C \times 3C$ matrix. Again, a sanity check gives that the full equivariant layer is then parameterized by $C^2 + 2C^2 + (3C)^2 = 12C^2$ values, which is the same as the group convolution discussed in Section 3.3.

We visualize the weight structure in Figure 7a.

N.3 IMPLEMENTATION

We implement a version of the Platonic Transformer with tetrahedral equivariance and all linear layers (i.e. in the MLP and projections in multi-head attention) in the Fourier domain. We transform back to the spatial domain at each non-linearity and at the RoPE-attention layers and to the Fourier domain after these layers. This transforming back-and-forth incurs an overhead that goes to zero as the hidden dimension increases (since it is just the 12×12 matrix Q applied to each channel C), however it is non-negligible at low-medium number of hidden dimensions, because it involves non-contiguous reshapes.

1674 The maximum FLOP saving that can be obtained from changing a linear layer to be in the Fourier
 1675 domain is going from $(12C)^2 = 144C^2$ operations to $C^2 + (2C)^2 + 3 \cdot (3C)^2 = 32C^2$, i.e. a saving
 1676 of 4.5 times. However, in order to make the implementation more efficient in pure PyTorch, we opt
 1677 to implement the mappings for ρ_1 and ρ_2 as one single $3C \times 3C$ matrix, enabling the whole linear
 1678 layer to be implemented as a batched matrix multiplication with four $3C \times 3C$ weight matrices, as
 1679 illustrated in Figure 7b. This batched implementation uses $4 \cdot (3C)^2 = 36C^2$ operations, yielding a
 1680 maximum potential compute saving of 4 times.

1681 N.4 THROUGHPUT BENCHMARKING

1682 We benchmark the training time per epoch on a subset of 20k molecules on the OMol25 task, using
 1683 PyTorch’s `torch.compile`. These timing runs are on a single NVIDIA RTX6000 GPU. We
 1684 keep all hyperparameters constant as in the main experiments, except for varying the number of
 1685 hidden dimensions. The results are presented in Table 9. It is clear that as we increase the number
 1686 of hidden dimensions, a Fourier implementation starts paying off more and more. Notably, since
 1687 the standard spatial implementation is equal to non-equivariant Transformers in computational cost,
 1688 the efficiency improvement of the Fourier implementation is a benefit of equivariant architectures
 1689 over non-equivariant ones. We emphasize that our Fourier implementation is not well-optimized, so
 1690 further throughput improvements should be available.

1691 Table 9: Training times per epoch (seconds) on a subset of OMol25 with 20k examples. We compare
 1692 a tetrahedral Platonic Transformer implemented in the spatial domain with one implemented in the
 1693 Fourier domain.

1698 Implementation	1699 Hidden dimension					
	576	864	1152	1440	1728	2016
Spatial (standard)	18	23	29	40	49	63
Fourier	19	22	27	32	38	45

1703 O DISCLOSURE OF LLM USAGE

1704 We declare that the use of LLMs for writing this paper was limited to general-purpose writing
 1705 assistance. Specifically, we used them only to polish the wording of text sections and in no way to
 1706 generate the research ideas or technical results and proofs presented in this paper.