GEOMETRIC MEAN POOLING FOR EQUAL-WEIGHT MULTIPLICATIVE COARSE-GRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

As an alternative to the additive and extremal biases of average and max pooling, we introduce Geometric Mean Pooling (GMP), a signed pooling operator that combines the product of feature signs with the geometric mean of feature magnitudes. Motivated by local-to-global composition in quantum many-body physics, GMP retains both joint sign information and a characteristic multiplicative scale without introducing learnable pooling parameters. We show that non-overlapping hierarchical GMP preserves the corresponding global multiplicative statistic and evaluate it on synthetic sequence tasks, iterative coarse-graining, image classification, and molecular lipophilicity regression. On the synthetic tasks, GMP recovers product-based signals more accurately than average and max pooling and maintains predictive performance under the tested levels of multiplicative input noise. On image and molecular data, however, its effectiveness depends on the representation, target parameterization, and placement of local and global pooling. These results position GMP as a complementary, regime-dependent inductive bias for tasks in which equal-weight multiplicative composition is plausible, rather than as a universal replacement for standard pooling operators.

1 INTRODUCTION

Pooling is a fundamental form of coarse-graining in convolutional neural networks (CNNs): it reduces spatial or sequence resolution while retaining features useful for downstream prediction. Classical CNN architectures established local max pooling as a source of spatial selectivity and translation tolerance (LeCun et al., 1998; Krizhevsky et al., 2012), while average pooling provides a complementary aggregation that preserves local mean responses. Global average pooling later became a standard way to replace fully connected layers with a spatially aggregated representation, reducing parameter count and encouraging correspondence between feature maps and class-level predictions (Lin et al., 2014; He et al., 2016). Modern surveys and learnable pooling methods have extended this basic design space to generalized means, learned norms, attention-based aggregation, and higher-order feature interactions (Tao et al., 2022; Gulcehre et al., 2013; Cui et al., 2017). These operators are effective when the target is controlled by additive statistics, salient responses, or learned weighted combinations of local features. They are less naturally matched to signals encoded in the joint product of many comparable features.

Multiplicative interactions provide a distinct alternative to additive aggregation. Product units and multiplicative neural networks were introduced to represent interactions that cannot be expressed efficiently by purely additive units (Durbin & Rumelhart, 1989; Miikkulainen, 1996), and multiplicative gates remain central to recurrent and attention-based architectures (Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017). Similar product-like structure appears in scientific settings in which a global quantity is assembled from local factors, including molecular properties, variational wavefunctions, and statistical-mechanical order parameters. In the present work, we focus on the more specific equal-weight case in which the target depends on $\prod_{i=1}^{N} x_i$, or, for positive magnitudes, equivalently on the sum $\sum_i \log |x_i|$. The individual feature marginals may then be identical across classes even though their joint products differ. In such a setting, average and max pooling can discard the relevant dependence structure, whereas a multiplicative pooling operator can preserve the product of feature signs and the average of their log-magnitudes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Theoretically, quantum many-body states provide additional physical motivation for studying multiplicative composition. In the non-interacting limit, single-particle eigenmodes provide independent degrees of freedom, and a many-body occupation state is generated by applying creation operators to the vacuum; for example, $|\Psi_{k_1,k_2}\rangle = a^\dagger_{k_1} a^\dagger_{k_2} |0\rangle$, where $a^\dagger_k$ creates a particle in eigenstate k. Thus, the occupation-number basis forms a product basis in Fock space, whereas interactions mix these configurations and generate superpositions and entanglement (Fetter & Walecka, 1971). For a fermionic occupation state built from single-particle orbitals, the coordinate-space wavefunction is a Slater determinant: an antisymmetrized sum of products of one-particle orbitals (Slater, 1929; Dirac, 1926; Foulkes et al., 2001). In neural-network quantum-state representations, this determinant is commonly retained as an explicit antisymmetric component, while neural networks parameterize the orbitals and correlation factors (Slater, 1929; Dirac, 1926; Pfau et al., 2020; Hermann et al., 2020). Valence-bond-solid (VBS) states provide a related example of structured multiplicative composition, with amplitudes organized from local singlet-bond factors and a superposition over compatible bond configurations (Anderson, 1973; Affleck et al., 1987). GMP plays a central role in the construction of exact neural-network quantum states for VBS states (Wu et al., 2026b).

With the observations mentioned above, GMP, as a parameter-free operator that averages feature log-magnitudes while retaining sign parity, provides a multiplicative complement to average and max pooling. We study global GMP, which summarizes full feature maps or sequences, and local GMP, which preserves blockwise multiplicative summaries for subsequent processing. GMP outperforms average and max pooling applied directly to inputs on synthetic signed-product classification tasks; image classification and molecular regression examine its dependence on learned representations, pooling placement, and target parameterization. Together, these studies connect the preservation of multiplicative observables across scales to pooling choice as a structural prior.

Contributions **(1) A signed multiplicative pooling primitive.** We define a pooling operator that combines sign parity with an equal-weight geometric mean of magnitudes for local and global aggregation, and distinguish it from unsigned geometric pooling and learned arithmetic modules. **(2) A coarse-graining analysis.** We establish hierarchical consistency for complete, equal-sized, non-overlapping groups with a shared clamp, and characterize how input signs, magnitudes, and activation choices affect the operator. **(3) Controlled evaluation of applicability.** We organize synthetic tasks around the statistics preserved by GMP, and use image and molecular experiments to examine its dependence on representation, activation, and pooling placement. Matched comparisons with average and max pooling isolate the role of the pooling primitive.

2 Related Work

Generalized means and pooling. Pooling methods include fixed average and max operators, learned norms, generalized means, and higher-order feature summaries (Tao et al., 2022; Gulcehre et al., 2013; Cui et al., 2017). Generalized Mean pooling (GeM), used in image retrieval (Radenović et al., 2018; Li et al., 2016), is defined for positive inputs by

$$\mathrm{GeM}_p(\mathbf{x}) = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^p \right)^{1/p}.$$

Its limit as $p \to 0$ is the geometric mean. Weighted geometric pooling (Weng et al., 2013), alpha-integration pooling (Eom & Choi, 2018), attention-aware GeM (Gu et al., 2018), and groupwise GeM (Ko et al., 2022) further explore this family.

GenAgg (Kortvelesy et al., 2023) provides a directly relevant framework based on generalized f-means, $f^{-1}(n^{-1}\sum_i f(x_i))$, with additional parameters extending the aggregation family. Its geometric-magnitude construction uses log-magnitudes. GMP builds on this established connection between geometric magnitudes and averaging in transformed coordinates. Its magnitude component inherits the equal-sized hierarchical consistency of quasi-arithmetic means. GMP fixes the magnitude transform and adds the product of input signs; for strictly positive inputs, GMP clamps the inputs from below, averages their logarithms, and exponentiates the result.

Learned multiplication and set representations. Neural Arithmetic Units include the Neural Multiplication Unit (NMU), which learns products over subsets of inputs (Madsen & Johansen,

2020). Neural Power Units (NPU) learn power functions with a treatment of negative inputs through complex arithmetic (Heim et al., 2020). These methods address learnable arithmetic structure, whereas GMP applies a prescribed parity rule and equal log-magnitude weights. GMP's sign-parity convention provides a specific real-valued composition rule alongside these learned power functions. Deep Sets (Zaheer et al., 2017) studies invariant representations built from learned elementwise transformations and sum aggregation. This separates two complementary design choices: selecting the aggregation statistic and learning the representation supplied to it. Our experiments focus on the first choice through matched pooling substitutions.

Geometry and physical coarse-graining. Geometric pooling on graphs and meshes concerns the geometry of the domain (Xu & Liu, 2024; Milano et al., 2020; Bianchi et al., 2025; Seong et al., 2018); geometric network renormalization likewise studies structural coarse-graining (Zheng et al., 2024). Here "geometric" refers to a mean of feature magnitudes. Connections between neural pooling, renormalization, and tensor networks motivate a block-coarse-graining interpretation (Kadanoff, 1966; Wilson, 1971; Hallam & Whitfield, 2018; Tang & Klette, 2016; Wu et al., 2022; 2025). We use this interpretation to study which statistics are preserved as local summaries are composed across scales.

3 METHODOLOGY

3.1 THE GEOMETRIC MEAN POOLING PRIMITIVE

We define the signed geometric mean pooling (GMP) over a window $W = \{x_1, \dots, x_k\}$ as

$$\mathrm{GMP}(W) = \left(\prod_{i=1}^{k} \mathrm{sign}(x_i)\right) \exp\left(\frac{1}{k}\sum_{i=1}^{k} \log\big(\max(|x_i|, \varepsilon)\big)\right), \tag{1}$$

where $\varepsilon > 0$ is a lower bound on magnitudes inside the logarithm and $\mathrm{sign}(0) = 0$. The implementation defaults are 10^{-6} in 1D and 10^{-12} in 2D; the hierarchical statement below assumes the same value at every level. The first factor records sign parity for nonzero inputs, and the second computes a clamped geometric magnitude. Log-space evaluation avoids forming the full magnitude product. Appendix A characterizes gradients and behavior near zero.

For nonzero inputs with an inactive clamp, GMP is related to the product decomposition

$$\prod_{i=1}^{k} x_i = \left(\prod_{i=1}^{k} \mathrm{sign}(x_i)\right) \exp\left(\sum_{i=1}^{k} \log|x_i|\right).$$

Thus, GMP preserves the joint sign and an equal-weight multiplicative scale. For known k and an inactive clamp, the product is recovered as $\mathrm{sign}(\mathrm{GMP})|\mathrm{GMP}|^k$. The normalization gives GMP a characteristic magnitude scale; reconstruction of the full product additionally uses the window size.

In the global regime, the window is the entire sequence, so $k = N$, and GMP reduces to a single equal-weight multiplicative summary of the whole input. In the local regime, GMP is applied blockwise with non-overlapping windows of size k and stride k:

$$\mathrm{GMP}_{\mathrm{local}}(\mathbf{x}) = [\mathrm{GMP}(W_1), \dots, \mathrm{GMP}(W_m)], \qquad m = \left\lfloor \frac{N}{k} \right\rfloor. \tag{2}$$

Each block is replaced by its signed multiplicative summary. If k does not divide N, this definition omits the trailing entries; the hierarchical identity applies only when the blocks cover all inputs. For 2D inputs, Eq. 1 is applied independently to each channel over an $r \times s$ spatial window, with $k = rs$.

RENORMALIZATION-GROUP PERSPECTIVE AND ALGEBRAIC CONTRAST

From a machine-learning perspective, a pooling layer replaces each local group of activations with a lower-resolution feature. Repeated application therefore composes local summaries into progressively coarser representations, which can be formalized as a coarse-graining map: each block is replaced by a single summary variable. In the language of Kadanoff block-spin RG (Kadanoff, 1966; Wilson, 1971), a lattice of microscopic variables $\{x_i\}_{i=1}^{N}$ is partitioned into non-overlapping

blocks B_j of size b, and each block is replaced by a single block variable $x'_j = \mathcal{R}_b(B_j)$. Repeated application generates a flow

$$x^{(0)} \xrightarrow{\mathcal{R}_b} x^{(1)} \xrightarrow{\mathcal{R}_b} x^{(2)} \xrightarrow{\mathcal{R}_b} \cdots \xrightarrow{\mathcal{R}_b} x^{(n)}, \qquad N_n = N/b^n.$$

An order parameter is a fixed point of this flow if its value is preserved across scales; in other words, repeated blockwise coarse-graining yields the same statistic as a single global summary. This is the natural RG language for distinguishing pooling operators.

For average pooling, the block map is the additive block-spin transform $\mathcal{R}_b^{\mathrm{avg}}(B) = \frac{1}{b}\sum_{i\in B} x_i$, and for max pooling it is the extremal order-statistic map $\mathcal{R}_b^{\max}(B) = \max_{i\in B} x_i$. These are natural RG maps for linear or extremal observables, but they are not fixed points for equal-weight multiplicative statistics. By contrast, GMP defines the block map $\mathcal{R}_b^{\mathrm{GMP}}(B) = \big(\prod_{i\in B}\operatorname{sign}(x_i)\big)\exp\big(\frac{1}{b}\sum_{i\in B}\log\big(\max(|x_i|,\varepsilon)\big)\big)$. Under the change of variables $u_i = \log\big(\max(|x_i|,\varepsilon)\big)$, the magnitude part of GMP becomes an arithmetic average in log-magnitude space:

$$\log|\mathcal{R}_b^{\mathrm{GMP}}(B)| = \frac{1}{b}\sum_{i\in B} u_i.$$

Hence GMP is algebraically conjugate to additive averaging in log-space, and repeated non-overlapping block pooling preserves the global multiplicative statistic exactly (proof in Appendix B). In this precise sense, GMP is a fixed-point RG map for equal-weight multiplicative order parameters.

When the target depends on a product of many features, average and max pooling may discard information needed to recover the target. In Case A of the experiments below, GMP directly retains the sign parity that determines the class label, whereas the tested average- and max-pooling classifiers perform near chance in the independent-input setting.

3.2 Local and Global Placement

Global GMP summarizes a full sequence or feature map and is structurally aligned with a global equal-weight multiplicative statistic. Local GMP preserves blockwise statistics for downstream processing and is aligned with tasks in which the relevant factors occur within those blocks. These two placements let the pooling scale reflect the spatial or sequential organization of the signal.

The pooling inputs determine the role of the sign component. With strictly positive inputs, GMP equals the geometric mean of clamped inputs. With ReLU inputs, any zero in a window forces its signed-GMP output to zero, and the implemented backward pass gives zero derivatives for all inputs in that window. Thus, activation choice affects both the information available to pooling and its optimization behavior.

4 Experiments

The synthetic tasks examine prescribed sign and magnitude statistics; the application experiments examine pooling within learned representations.

4.1 Synthetic Sequence Classification

Case A (global signed-product classification). For each example, we draw a clean sequence $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ of length N and set the label $y = \mathbf{1}\{\prod_{i=1}^{N} x_i < 0\}$, with each x_i being independent identically distributed (i.i.d.). Equivalently, the two classes differ only in the parity of negative entries: class 0 has an even number of negative entries and class 1 an odd number. For this independent-input setting ($N > 1$), the class prior is balanced and every individual coordinate has the same marginal distribution in both classes. We compare global max, average, and GMP, each followed by a linear classifier: $\mathrm{Pool}_{\mathrm{global}} \to \mathrm{FC}(1,2)$, where FC denotes a fully connected layer mapping. No convolution, normalization, or pointwise activation is used, so that the pooling operator acts directly on the variables carrying the product structure. Across the reported single-seed sweeps over sequence length $N \in \{16, 32, 64, 128\}$ and adjacent-site correlation $\rho \in \{0, 0.3, 0.5, 0.7, 0.9\}$, GMP achieves 100% accuracy. Average and max pooling remain near chance for independent inputs and both reach 60.4% accuracy at $\rho = 0.9$. Details and full results appear in Appendix C, Table 3.

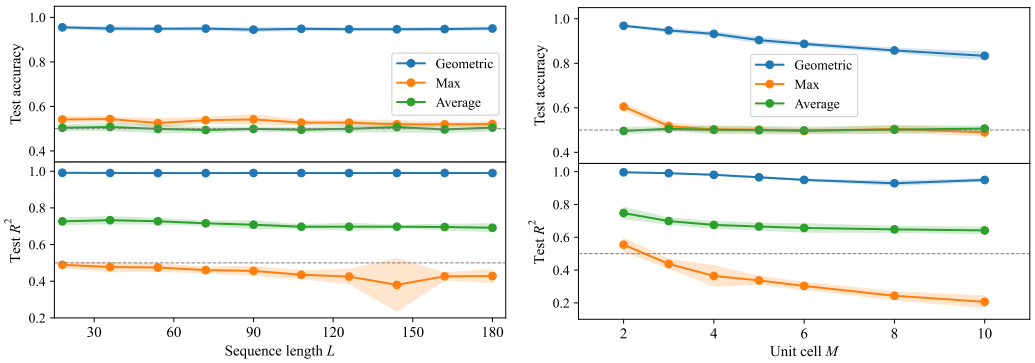

Figure 1: Sequence-length and cell-size sweeps for Case B classification (top) and R^2 regression (bottom). Left: varying sequence length with cell size 3. Right: varying cell size with sequence length 120. Clean classification inputs are standard Gaussian; clean regression inputs are lognormal with standard Gaussian logarithms. Targets are computed before adding Gaussian input noise with standard deviation 0.05. Error bars denote the standard deviation across ten random seeds.

Case B (local signed-product-sum classification). To test local multiplicative coarse-graining, we partition a clean sequence $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ into non-overlapping cells C_j of size k. Each cell contributes the signed geometric mean

$$g_j = \left(\prod_{i \in C_j} \operatorname{sign}(x_i) \right) \exp\left(\frac{1}{k} \sum_{i \in C_j} \log(\max\{|x_i|, \varepsilon\}) \right),$$

and the sample-level statistic is $S = \sum_{j=1}^{N/k} g_j$, $y = \mathbf{1}\{S < 0\}$. Thus the label depends on a sum of local, equal-weight multiplicative statistics. The classifier receives a noisy observation $\tilde{x}_i = x_i + \eta_i$, where $\eta_i \sim \mathcal{N}(0, 0.05^2)$ independently; labels are always computed from the clean sequence.

For every pooling primitive, we use the matched architecture

$$\text{Pool}_{\text{local}}(k, \text{stride} = k) \;\to\; \text{Flatten} \;\to\; \text{FC}(N/k, 2).$$

The local windows are aligned exactly with the generative cells. As in Case A, there are no convolutional layers, normalizations, or activations before pooling. This deliberately minimal setting isolates the information preserved by the pooling primitive: on clean inputs with matching numerical conventions, GMP produces the local statistics g_j directly, after which the linear classifier learns their sum. Noisy observations test how accurately these local summaries retain the clean class signal.

The classification panels of Fig. 1 show a consistent advantage for GMP over average and max pooling in the aligned settings. Accuracy remains high across the sequence-length sweep and declines as the cell size grows. The two sweeps illustrate how the operator's multiplicative prior interacts with the number and size of local groups under input noise.

Impact of Sign Information. To quantify the effect of sign information on classification accuracy, we perform an ablation study comparing signed and unsigned geometric-mean pooling (Table 1). We first embed each input sequence into a higher-dimensional space to increase model expressivity, using the architecture

$$\text{Conv1d}(1, d_{\text{embed}}, k = 1) \;\to\; \text{Pool} \;\to\; \text{FC}(d_{\text{embed}}, 2),$$

with $d_{\text{embed}} = 32$. We evaluate both Case A and Case B under the corresponding global and local pooling settings. The only difference between the pooling variants is whether the geometric mean is computed with or without the sign-parity term. Removing the sign-parity term reduces accuracy to approximately chance level in both tasks, demonstrating its importance in these settings.

Table 1: Ablation of the sign-parity term in geometric mean pooling. Results are reported as mean $\pm$ standard deviation across 10 random seeds. The sequence length is 120, and the cell size is 3 for Case B.

Setting	Accuracy
Case A, signed	0.9330 ± 0.0659
Case A, unsigned	0.5107 ± 0.0165
Case B, signed	0.9856 ± 0.0047
Case B, unsigned	0.5072 ± 0.0136

4.2 Synthetic Magnitude Regression and Iterative Pooling

The synthetic regression tasks use MSE loss and a linear output layer. Lognormal clean inputs provide a controlled setting for studying multiplicative magnitudes. Local aggregation, learned embeddings, and input perturbations then probe how this structure is carried into the prediction.

R1 (Global geometric-mean regression). The clean task draws $x_i \sim \mathrm{lognormal}(0, 1)$ independently and sets $y = \exp\big(N^{-1} \sum_i \log x_i\big)$. The model uses a learned pointwise $\mathrm{Conv1d}(1, d_{\mathrm{embed}}, k = 1)$, followed by variable global pooling and a linear regression head, with $d_{\mathrm{embed}} = 1$ for R1. The geometric mean of the raw inputs provides the exact target statistic when the clamp is inactive. The learned affine embedding and regression head allow the model to adapt this summary during training.

For $N \in \{16, 32, 64, 128\}$ in the reported single-seed experiment, GMP achieves $R^2 = 0.993$–1.000, compared with 0.609–0.638 for average pooling and 0.019–0.137 for max pooling. These results support the correspondence between geometric aggregation and the multiplicative target; average pooling also retains predictive information through correlations between arithmetic and geometric magnitudes. Full results appear in Appendix C, Table 4.

R2 (Local cell log-mean regression): We use i.i.d. lognormal features, $x_i \sim \mathrm{lognormal}(0, 1)$, with cell size $c \in \{2, 3, 4, 5, 6, 8, 10\}$ and sequence length $N \in \{18, 36, 54, 72, 90, 108, 126, 144, 162, 180\}$ (fixed $c = 3$) or $N = 120$ (varying c). Each cell contributes $s_j = \exp\Big(\frac{1}{c} \sum_{i \in \mathrm{cell}_j} \log |x_i|\Big)$, and the target $y = \frac{1}{n_{\mathrm{cell}}} \sum_j s_j$ is computed from the *clean* sequence. The architecture is local pooling (kernel and stride c), flattening, and $\mathrm{FC}(n_{\mathrm{cell}}, 32) \to \mathrm{ReLU} \to \mathrm{FC}(32, 1)$. Pooling is selected from GMP, average, and max and acts directly on the noisy sequence, without a preceding convolutional embedding. The sweeps appear in Fig. 1 (bottom panels). Observations are perturbed as $\tilde{x}_i = x_i + \eta_i$, with $\eta_i \sim \mathcal{N}(0, 0.05^2)$.

R3 (Noise robustness): Clean lognormal features are corrupted by multiplicative noise applied to every feature, followed by additive noise: $\tilde{x} = x \cdot \exp(\epsilon_{\mathrm{mul}}) + \epsilon_{\mathrm{add}}$, $\epsilon_{\mathrm{mul}} \sim \mathcal{N}(0, \sigma^2_{\mathrm{mul}})$, $\epsilon_{\mathrm{add}} \sim \mathcal{N}(0, \sigma^2_{\mathrm{add}})$. The target is the geometric mean of the clean features. The model uses the same architecture as R1, but with $d_{\mathrm{embed}} = 32$ to increase the capacity of the learned representation before global pooling.

Figure 2 compares responses to multiplicative and additive perturbations. Multiplicative noise shifts raw log-magnitudes additively, directly matching the coordinates used by GMP. Additive noise probes a different perturbation and can also change input signs. The sweep examines these two effects within the same prediction task.

R4 (Iterative preservation of a multiplicative statistic). Clean lognormal sequences of length $L = 256$ are perturbed by multiplying a randomly selected fraction $p = 25\%$ of entries by $\exp(\eta)$, with $\eta \sim \mathcal{N}(0, 1)$. Repeated pooling uses kernel and stride 2, reducing 256 entries to one in eight steps. At every step, the evaluator takes a global geometric-mean readout of the remaining entries for *all three* pooling methods and compares it with the clean global geometric mean using MAE.

Figure 3 illustrates the connection between hierarchical consistency and preservation of a multiplicative observable. The GMP curve remains approximately constant across successive pooling steps, whereas average and max pooling change the common geometric readout. This behavior reflects the statistic preserved by each block transformation: GMP composes geometric summaries, average pooling composes arithmetic means, and max pooling composes extrema.

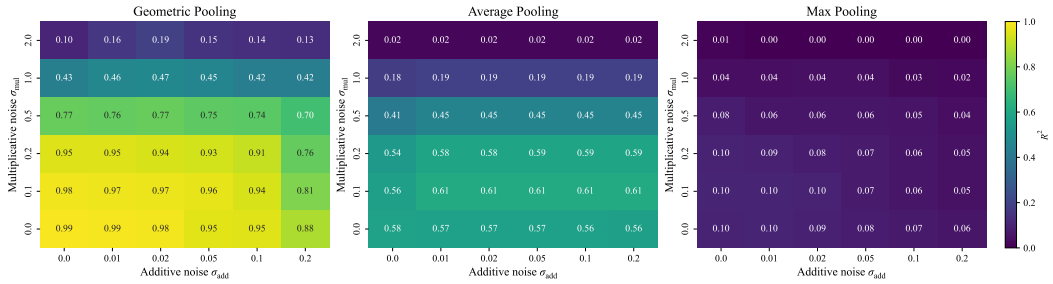

Figure 2: R3: global regression under multiplicative and additive input noise, $\tilde{x} = x \exp(\epsilon_{\text{mul}}) + \epsilon_{\text{add}}$, with $\log x \sim \mathcal{N}(0, 1)$ and independent Gaussian perturbations of standard deviations σ_{mul} and σ_{add}. Over the tested noise ranges, GMP maintains higher predictive performance than max pooling, including at low noise levels.

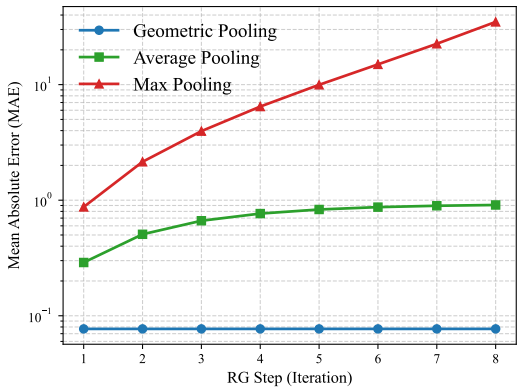

Figure 3: R4: preservation of a multiplicative observable across pooling levels. MAE is evaluated against the clean geometric-mean target ($\sigma = 1$, $p = 25\%$, $L = 256$), using a global geometric readout after each local pooling step for all methods. GMP maintains an approximately constant readout error across the hierarchy.

4.3 Image Classification and Activation Compatibility

We evaluate GMP on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky, 2009) using convolutional classifiers with matched channel dimensions, optimizers, and training procedures. The experiments isolate three different placements of the variable pooling primitive: global aggregation, repeated local spatial reduction, and synchronized local-plus-global pooling. For each configuration, we compare GMP, max pooling, and average pooling.

The three configurations specify where the variable pooling operator is used:

Configuration	Local pooling	Global pooling
Global	fixed max	variable
Local	variable	fixed max
Synchronized	variable	same as local

The variable operator is selected from $\{\text{GMP}, \text{Max}, \text{Avg}\}$. Thus, the Global configuration isolates the final spatial summary, the Local configuration tests repeated local coarse-graining, and the Synchronized configuration applies the same pooling operator at both levels. All backbones use convolution–ReLU blocks, with matched architectures and training procedures within each dataset. Full architecture details appear in Appendix D.

Figure 4 shows that local GMP remains competitive with max and average pooling on MNIST and Fashion-MNIST, whereas global GMP performs less consistently across the tested settings. The performance gap is larger on CIFAR-10. The sensitivity of signed GMP to exact zeros after ReLU offers

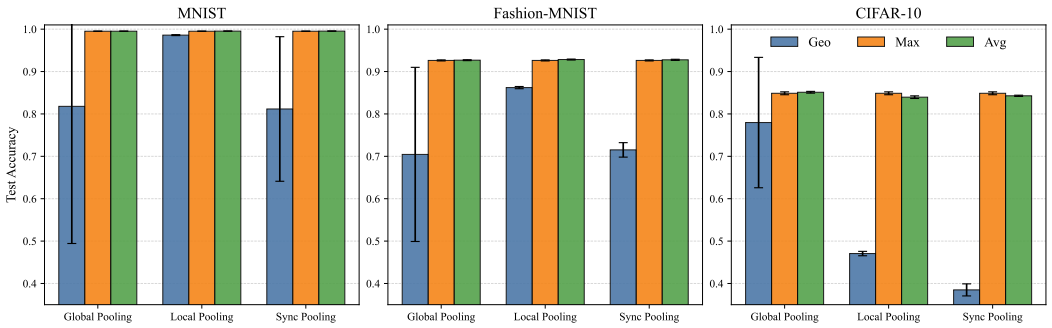

Figure 4: Validation accuracy on MNIST, Fashion-MNIST, and CIFAR-10 across ten random seeds. Each panel reports the three pooling primitives (GMP, max, and average) under three architectures: variable global pooling with fixed intermediate max pooling, variable local pooling with fixed global max pooling, and synchronized local-plus-global pooling. Local GMP is generally more competitive than global GMP on MNIST and Fashion-MNIST, whereas the gap between GMP and the average- and max-pooling baselines increases on CIFAR-10. Error bars denote standard deviation across 10 seeds. All models are trained with cross-entropy loss and Adam (Kingma & Ba, 2015) using batch size 128 and learning rate 10^{-3}.

a possible explanation for the weaker global results. The results therefore support a regime-specific interpretation of GMP: it is a useful structural prior when multiplicative composition is plausible, but it is not generally interchangeable with additive pooling for generic image classification.

4.4 Lipophilicity Regression

We evaluate on the Lipophilicity regression task from the MoleculeNet molecular benchmark suite (Wu et al., 2018), specifically version 1 (V1), curated from ChEMBL with 4,200 small organic molecules. Molecules are partitioned by a Bemis–Murcko scaffold split (Bemis & Murcko, 1996), with approximately 80% of scaffolds assigned to training and the remaining scaffolds assigned to testing, so the test set contains molecular scaffolds absent from training. The dataset label is a log-scale lipophilicity value, $y = \log D$, the experimental octanol/water distribution coefficient at pH 7.4 (range approximately $[-1.5, 4.5]$). To restore the true multiplicative structure of the molecular property, we transform the regression target as $z = \exp(y)$ before optimization and evaluation, so the reported RMSE and R^2 are measured on the z scale (range approximately $[0.2, 90]$) rather than the $\log D$ scale.

Each molecule is represented by a 2,048-bit Morgan fingerprint (FP) with radius 2 (ECFP4) (Rogers et al., 2010). Because the raw fingerprint is fixed, high-dimensional, and sparse, our 2D CNN maps it through a fully connected projection layer, $\mathbb{R}^{2048} \to \mathbb{R}^{1024}$ followed by ReLU, giving the network learnable capacity to re-express the fingerprint into a denser representation rather than operating directly on raw binary bits. The projected vector is reshaped into a single-channel 32×32 latent feature map, to which a softplus activation is applied before three convolutional blocks, each consisting of a 3×3 convolution (channel widths $1 \to 32 \to 64 \to 128$), the same activation, and a 2×2 local pooling operator with stride 2 (reducing the spatial resolution $32 \times 32 \to 16 \times 16 \to 8 \times 8 \to 4 \times 4$). The resulting 128-channel feature map is reduced by a global pooling operator to a 128-dimensional vector, which is passed to a two-layer regression head, $\mathrm{FC}(128, 32) \to \mathrm{ReLU} \to \mathrm{FC}(32, 1)$. For each experiment, the local and global pooling operators are independently selected from geometric, max, and average pooling. The learned 32×32 layout supplies latent neighborhoods for convolutional processing, with locality defined by the representation.

We compare the 2D CNN against four baselines trained on the same scaffold split, summarized in Table 2: Morgan fingerprint features with XGBoost (Chen & Guestrin, 2016), a Random Forest regressor (Breiman, 2001), and a fully connected network, as well as a learned SMILES (Weininger, 1988) token embedding passed through the same fully connected head. The three Morgan-fingerprint baselines achieve similar performance and outperform the tested SMILES-embedding baseline, suggesting that Morgan fingerprints provide a more effective representation in this experimental setting.

Table 2: Lipophilicity regression performance on the original logarithmic label scale, using a scaffold split.

Baseline	**RMSE**	R^2
Morgan FP + XGBoost	0.8870	0.4631
Morgan FP + Random Forest	0.8718	0.4813
Morgan FP + FC	0.8752	0.4773
SMILES Embedding + FC	1.1596	0.0824

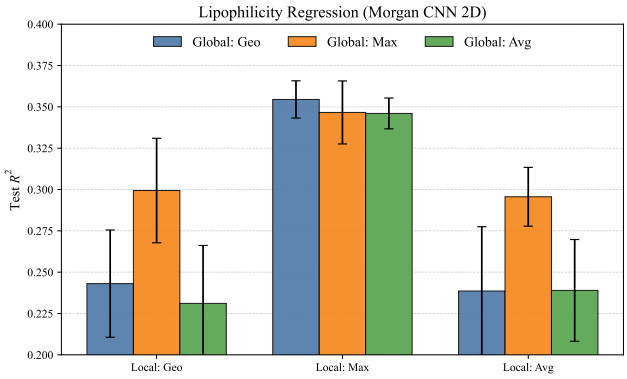

Figure 5: Test R^2 on the auxiliary target $z = \exp(y)$ for the Morgan-fingerprint 2D CNN under all combinations of local and global pooling operators. Each molecule is represented by a radius-2, 2,048-bit Morgan fingerprint, projected to a learned 32×32 latent feature map, and processed by a three-block convolutional hierarchy. Values and error bars denote the mean and standard deviation over ten random seeds.

Fig. 5 shows the R^2 performance of the Morgan FP 2D CNN under all combinations of local and global pooling operators. Global GMP outperforms max and average pooling, while local GMP becomes detrimental when combined with global max pooling. The best pooling configuration achieves $R^2 = 0.3545$ on the auxiliary target z. Because the baselines report results on the log scale of z, those values provide an indirect comparison with this experiment. The results overall highlight the importance of matching the pooling strategy to the underlying algebraic structure of the target property. As shown in Appendix E, training the same architecture on the original label scale changes the ranking, with max pooling achieving the strongest performance.

5 Conclusion

We introduced GMP, a parameter-free pooling operator that combines sign parity with equal-weight geometric magnitudes. It preserves the corresponding global statistic under complete, equal-sized, non-overlapping hierarchical pooling with a shared clamp and no intervening transformations. Synthetic experiments demonstrate its utility for prescribed multiplicative signals, while image and molecular experiments show that its effectiveness depends on representation, activation, pooling placement, and target parameterization. GMP therefore provides a structural prior for sign-sensitive or equal-weight multiplicative tasks. For positive targets, predicting their logarithms with additive aggregation offers an alternative. Extending this framework to complex-valued representations (Trabelsi et al., 2017), developing learnable hybrid operators, and testing them on physical (Wu et al., 2026a; Carleo & Troyer, 2017; Pfau et al., 2020), chemical (Gilmer et al., 2017; Schütt et al., 2017; Yang et al., 2019), and other scientific datasets are important directions for future work.

AI use statement

In this work, we used generative AI tools for assisting in the writing of proofs, implementing methods, cleaning and reformatting dataset, designing or providing feedback on research methodology or experiments, generating synthetic data sets. We have not used generative AI tools for helping

develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, supporting qualitative and thematic data analysis, interpreting results. Assisting with translation is not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, summarizing or analysing existing literature, brainstorming, sourcing/searching for information, editing a research paper to improve readability, identifying relevant literature, formating references. We have reviewed all AI-assisted work. We checked LLM-generated method implementations are correct and data were properly handled, and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

The results presented in this paper are readily reproducible based on the descriptions, algorithms, and code provided in the main text, appendix. The codes are publicly available at `https://anonymous.4open.science/r/GeometricMeanPooling-731E/README.md`.

References

Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Physical Review Letters*, 59(7):799–802, 1987. doi: 10.1103/PhysRevLett.59.799.

Philip W. Anderson. Resonating valence bonds: A new kind of insulator. *Materials Research Bulletin*, 8(2):153–160, 1973. doi: 10.1016/0025-5408(73)90167-0.

Guy W. Bemis and Mark A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928.

Filippo Maria Bianchi, Carlo Abate, Ivan Marisca, et al. Torch geometric pool: the pytorch library for pooling in graph neural networks. *arXiv:2512.12642*, 2025.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. doi: 10.1126/science.aag2302.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2930, 2017.

Paul A. M. Dirac. On the theory of quantum mechanics. *Proceedings of the Royal Society A*, 112 (762):661–677, 1926. doi: 10.1098/rspa.1926.0133.

Richard Durbin and David E. Rumelhart. Product units: A computationally powerful extension to backpropagation networks. *Neural Computation*, 1(1):133–142, 1989. doi: 10.1162/neco.1989.1.1.133.

Hayoung Eom and Heeyoul Choi. Alpha-integration pooling for convolutional neural networks. *arXiv:1811.03436*, 2018.

Alexander L. Fetter and John Dirk Walecka. *Quantum Theory of Many-Particle Systems*. McGraw-Hill, New York, 1971.

W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73:33–83, Jan 2001. doi: 10.1103/RevModPhys.73.33. URL `https://link.aps.org/doi/10.1103/RevModPhys.73.33`.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning*, 70:1263–1272, 2017.

Yinzheng Gu, Chuanpeng Li, and Jinbin Xie. Attention-aware generalized mean pooling for image retrieval. *arXiv:1811.00202*, 2018.

Caglar Gulcehre, Kyunghyun Cho, Razvan Pascanu, and Yoshua Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. *arXiv:1311.1780*, November 2013. Comments: ECML/PKDD 2014.

Gareth Hallam and Peter Whitfield. Compact neural networks based on the multiscale entanglement renormalization ansatz (mera). In *Proceedings of the 25th Australian Pattern Recognition Conference (BMVC)*, pp. Paper 103, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Niklas Heim, Tomas Pevny, and Vasek Smidl. Neural power units. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/48e59000d7dfcf6c1d96ce4a603ed738-Abstract.html.

Jan Hermann, Zachary Schätzle, and Frank Noé. Ab initio quantum chemistry with neural-network wavefunctions. *Nature Chemistry*, 12:891–897, 2020. doi: 10.1038/s41557-020-0544-y.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

L. P. Kadanoff. Scaling laws for ising phase transitions or how everything goes wrong when the dimension is two. *Physics*, 1(5):263–272, 1966. doi: 10.1103/Physics.1.263.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Byungsoo Ko, Han-Gyu Kim, Byeongho Heo, Sangdoo Yun, Sanghyuk Chun, Geonmo Gu, and Wonjae Kim. Group generalized mean pooling for vision transformer. *arXiv:2212.04114*, 2022.

Ryan Kortvelesy, Steven Morad, and Amanda Prorok. Generalised f-mean aggregation for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/6c78ae0c1140902bf3a430b1725bcc4e-Abstract-Conference.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Teng Li, Z. Meng, Bingbing Ni, Jianbing Shen, and Meng Wang. Robust geometric ℓ_p-norm feature pooling for image classification and action recognition. *Image and Vision Computing*, 55:1–12, 2016.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representations*, 2014.

Andreas Madsen and Alexander Rosenberg Johansen. Neural arithmetic units. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2001.05016.

Risto Miikkulainen. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, 1996.

Francesco Milano, Antonio Loquercio, Antoni Rosinol, Davide Scaramuzza, and Luca Carlone. Primal-dual mesh convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

David Pfau, James S. Spencer, James Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3): 033429, 2020. doi: 10.1103/PhysRevResearch.2.033429.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. doi: 10.1109/TPAMI.2018.2846566.

David Rogers, Matthew Brown, and Matthias Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Kristof T. Schütt, Anders Arbab-Birkirkjær, et al. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in Neural Information Processing Systems*, 30, 2017.

Si-Baek Seong, Chongwon Pae, and Hae-Jeong Park. Geometric convolutional neural network for analyzing surface-based neuroimaging data. *Frontiers in Neuroinformatics*, 12:42, 2018. doi: 10.3389/fninf.2018.00042.

John C. Slater. The theory of complex spectra. *Physical Review*, 34(10):1293–1322, 1929. doi: 10.1103/PhysRev.34.1293.

Mengdan Tang and Reinhard Klette. Learning scale-variant and scale-invariant features for deep image classification. *arXiv:1602.01255*, February 2016.

Zhi Tao, Xiaoyu Chen, Huiling Li, Xinyu Yang, Yuncan Liu, and Xiaomin Zhang. Pooling operations in deep learning: From "invariable" to "variable". *BioMed Research International*, 2022: 4067581, 2022. doi: 10.1155/2022/4067581.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *arXiv:1705.09792*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Chaoqun Weng, Hongxing Wang, and Junsong Yuan. Learning weighted geometric pooling for image classification. In *2013 IEEE International Conference on Image Processing*, pp. 3713–3717. IEEE, 2013.

K. G. Wilson. Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture. *Physical Review B*, 4(9):3174–3183, 1971. doi: 10.1103/PhysRevB.4.3174.

Ang-Kun Wu, Daniel Bauernfeind, Xiaodong Cao, Sarang Gopalakrishnan, Kevin Ingersent, and J. H. Pixley. Aubry-andré anderson model: Magnetic impurities coupled to a fractal spectrum. *Phys. Rev. B*, 106:165123, Oct 2022. doi: 10.1103/PhysRevB.106.165123. URL `https://link.aps.org/doi/10.1103/PhysRevB.106.165123`.

Ang-Kun Wu, Benedikt Kloss, Wladislaw Krinitsin, Matthew T. Fishman, J. H. Pixley, and E. M. Stoudenmire. Disentangling interacting systems with fermionic gaussian circuits: Application to quantum impurity models. *Phys. Rev. B*, 111:035119, Jan 2025. doi: 10.1103/PhysRevB.111.035119. URL `https://link.aps.org/doi/10.1103/PhysRevB.111.035119`.

Ang-Kun Wu, Louis Primeau, Jingtao Zhang, Kai Sun, Yang Zhang, and Shi-Zeng Lin. Modeling quantum geometry for fractional chern insulators with unsupervised learning. *npj Comput. Mater.*, May 2026a. doi: 10.1038/s41524-026-02155-1. URL https://www.nature.com/articles/s41524-026-02155-1.

Ang-Kun Wu, Louis Primeau, Yixin Zhang, Jingtao Zhang, Adrian Del Maestro, and Yang Zhang. Compact spin-charge separated neural quantum states for valence-bond states. *arXiv:2606.17045*, 2026b.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018. doi: 10.1039/c7sc02664a.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Yutong Xu and Zhi Liu. Geometric pooling: Maintaining more representative information for graph node classification. *Neurocomputing*, 579:127439, 2024. doi: 10.1016/j.neucom.2024.127439.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, E. S. Jensen, et al. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Muhua Zheng, Simon L. Evans, Gilles Humair, Michael Small, and Tianhou Zhou. Geometric renormalization of weighted networks. *Communications Physics*, 7(1):97, March 2024. doi: 10.1038/s42005-024-01589-7.