Geometric algebra transformers for large 3D meshes via cross-attention

Julian Suk [1] **Pim de Haan** [2] **Baris Imre** [1] **Jelmer M. Wolterink** [1]

Abstract

Surface and volume meshes of 3D anatomical structures are widely used in biomedical engineering and medicine. The advent of machine learning enabled viable applications which come with the unique challenge of applying deep neural networks to large 3D meshes. In this work, we scale the recently introduced geometric algebra transformers (GATr) to meshes with hundreds of thousands of vertices by projection to a coarser set of vertices via cross-attention. The resulting neural network inherits GATr's equivariance under rotation, translation and reflection, which are desirable properties when dealing with 3D objects.

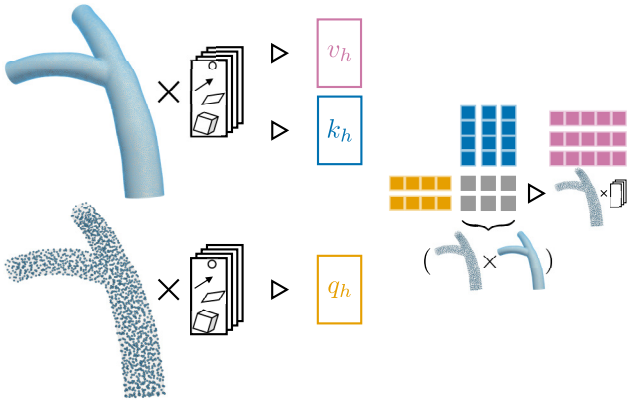

Figure 1. **Cross attention** between mesh vertices (keys k_h, values v_h) and sub-sampled points (queries q_h). Inputs are embedded in $\mathbf{G}(3,0,1)$. Attention scores determine a transformation from the fine to the coarse resolution. Output is in coarse resolution.

1. Introduction

The application of machine learning to 3D meshes of anatomical structures, e.g. in cardiovascular hemodynamics modelling, has been an ongoing area of research (Arzani et al., 2022; Li et al., 2021). Hemodynamics strongly depend on up- and downstream anatomy and their estimation requires global context across the mesh. Several previous works have focussed on graph neural networks (GNN) to estimate hemodynamic quantities (Suk et al., 2024a; 2023). However, GNNs are known to exhibit over-squashing (Alon & Yahav, 2021) and thus can be inefficient at accummulating the necessary receptive fields within large 3D meshes. Partly inspired by the success of large language models, transformers (Vaswani et al., 2017) have moved into the attention of the biomedical research community (Sarasua et al., 2021; Dahan et al., 2023; Suk et al., 2024b). Self-attention models global interactions between all mesh vertices and optimised implementations are available with linear memory complexity (Rabe & Staats, 2021).

Clifford and geometric algebras have been shown to be a viable inductive bias in neural networks for a multitude of tasks including fluid modelling (Brandstetter et al., 2023; Ruhe et al., 2023a;b; Brehmer et al., 2023). Ruhe et al. (2023b) and Brehmer et al. (2023) introduced neural network layers that are equivariant under transformations of the family of symmetry groups $\mathrm{Pin}(\alpha,\beta,\gamma)$ which capture all Euclidean symmetries (rotation, translation and reflection). Group equivariance has been shown to increase accuracy and data efficiency for cardiovascular hemodynamics estimation (Suk et al., 2024a; 2023).

3D meshes of patient anatomy should be understood as artefacts rather than features of the underlying data. The mesh connectivity and granularity are usually owed to the downstream application (e.g. fluid simulation) and rarely indicative of biomedical mechanisms. Large biomedical geometric algebra transformer (LaB-GATr) (Suk et al., 2024b) acknowledges this by introducing geometric tokenisation of the input mesh to a reduced sequence of tokens. This additionally enables control over the neural network's memory complexity which depends on the number of tokens. How-

*Equal contribution [1]Department of Applied Mathematics, Technical Medical Centre, University of Twente, Enschede, The Netherlands [2]Qualcomm AI Research, Qualcomm Technologies Netherlands B.V., (Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.). Correspondence to: Julian Suk <j.m.suk@utwente.nl>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

ever, LaB-GATr's tokenisation uses message passing layers which are by design restricted to accumulating local context. In this work we propose an extension to LaB-GATr by replacing its tokenisation by *projection*, via cross-attention, of the input mesh onto a coarse set of vertices, sub-sampled via farthest point sampling. This leads to matched performance on one and superior performance on another dataset in the context of cardiovascular hemodynamics estimation. Our implementation is available online.[1]

Related works Jaegle et al. (2022) proposed a transformer model that encodes arbitrary input data via cross-attention. This is similar to our model with the difference that we construct the input queries and thus "latent arrays" via sub-sampling of 3D geometry which links them to 3D space. Wang et al. (2023) propose a similar model that uses sub-sampling for the output queries but stil uses generic "latent arrays" that are not linked to 3D space. However, this geometric context might be relevant for our model because it operates in geometric algebra. In contrast to both above works, LaB-GATr and our extension are $\mathrm{E}(3)$-equivariant.

2. Geometric algebra

Clifford algebras, also known as geometric algebras, are an alternative to the ubiquitous linear algebra. Geometric algebra provides far simpler calculus when dealing with and manipulating geometric objects. While we will not formally introduce geometric algebra here[2], we will motivate basic geometric calculus. A geometric algebra $\mathbf{G}(\alpha, \beta, \gamma)$ is characterised by the signature (α, β, γ) of its bilinear form (used in the formal introduction). In this paper (compare (Suk et al., 2024b)), we use the projective geometric algebra $\mathbf{G}(3, 0, 1)$ in which a homogeneous coordinate is added to $\mathbb{R}^3$ in order represent translations in 3D as linear map. At the core of geometric algebra lies the introduction of an associative (but not commutative) geometric product of vectors y, z, simply denoted as yz. Given an orthogonal basis $\{e_i\}_i$, the following holds:

$$e_0 e_0 = 0, \quad e_i e_i = 1 \quad (i \neq 0), \quad e_i e_j = -e_j e_i \quad (i \neq j).$$

Elements $x \in \mathbf{G}(3, 0, 1)$, called multivectors, are composed of all possible linearly independent geometric products:

$$\begin{gathered} e_0 \\ e_0, e_1, e_2, e_3 \\ e_0e_1, e_0e_2, e_0e_3, e_1e_2, e_1e_3, e_2e_3 \\ e_0e_1e_2, e_0e_1e_3, e_0e_2e_3, e_1e_2e_3 \\ e_0e_1e_2e_3 \end{gathered}$$

which span a 16-dimensional vector space and can be equivalently written as

$$x = (\underbrace{x_s}_{\text{scalar}}, \underbrace{x_0, x_1, x_2, x_3}_{\text{vectors}}, \underbrace{x_{01}, x_{02}, x_{03}, x_{12}, x_{13}, x_{23}}_{\text{bivectors}}, \underbrace{x_{012}, x_{013}, x_{023}, x_{123}}_{\text{trivectors}}, \underbrace{x_{0123}}_{\text{pseudoscalar}}).$$

Brehmer et al. (2023) provide a look-up table of how to embed common geometric objects, such as points and planes, as multivectors.

3. Methods

3.1. Neural network architecture

We consider an extension to LaB-GATr (Suk et al., 2024b) by replacing its message-passing pooling module by cross-attention. We embed the n_{fine} mesh vertices as multivectors which are reduced to n_{coarse} point embeddings by the cross-attention. Afterwards, GATr (Brehmer et al., 2023) updates these with geometric self-attention. As in LaB-GATr, a learned interpolation block then recovers the original mesh resolution n_{fine}. To differentiate our model and LaB-GATr, we here denote ours LaB-GATr++.

3.2. Cross-attention

Given an embedding $X^{(0)}_{\text{fine}} \in \mathbb{R}^{n_{\text{fine}} \times c \times 16}$ of the mesh vertices, we compute an equidistantly spaced, coarse embedding $X^{(0)}_{\text{coarse}} \in \mathbb{R}^{n_{\text{coarse}} \times c \times 16}$ via farthest point sampling. We project the fine-scale embedding onto the coarse scale using a multi-head cross-attention block:

$$\begin{aligned} a_h^{(0)} &= \mathrm{Softmax}\left(\frac{q_h(X^{(0)}_{\text{coarse}}) k_h(X^{(0)}_{\text{fine}})^\mathsf{T}}{\sqrt{8c}}\right) v_h(X^{(0)}_{\text{fine}}) \\ A^{(0)} &= X^{(0)}_{\text{coarse}} + \xi\left(\underset{h}{\mathrm{Concat}}\ a_h^{(0)}\right) \\ X^{(1)} &= A^{(0)} + \phi(A^{(0)}). \end{aligned}$$

where q_h, k_h, v_h are vertex-wise permutation-equivariant layers consisting of layer normalisation composed with learned linear maps introduced in (Brehmer et al., 2023). Attention heads, indexed by h, are combined with a learned linear map ξ and feature updates are computed with a geometric nonlinear layer ϕ. For discussion of the scale factor $\sqrt{8c}$, see (Brehmer et al., 2023). The attention layer is visualised in Figure 1. The attention matrix $a_h^{(0)} \in \mathbb{R}^{n_{\text{coarse}} \times n_{\text{fine}}}$ constitutes a transformation map from the fine to the coarse resolution. Cross-attention blocks are equivariant under rotations, translations and reflections $\rho \in \mathrm{E}(3)$ of the input geometry as embedded in $X^{(0)}_{\text{fine}}$, i.e. $\rho X^{(0)}_{\text{fine}} \mapsto \rho X^{(1)}$.

[1] github.com/sukjulian/lab-gatr

[2] We refer the interested reader to (Brandstetter et al., 2023; Ruhe et al., 2023b;a; Brehmer et al., 2023)

3.3. Interpolation

Given a tensor $X^{(l)} \in \mathbb{R}^{n_{\text{coarse}} \times c \times 16}$ we recover the original mesh resolution with the following learned interpolation block (compare (Suk et al., 2024b)):

$$X^{(l+1)}|_v = \frac{\sum_p \lambda_{p,v}\, X^{(l)}|_p}{\sum_p \lambda_{p,v}}, \qquad \lambda_{p,v} \coloneqq \frac{1}{\|p - v\|_2^2 + \epsilon},$$
$$Y = \psi(X^{(l+1)}, X^{(0)}_{\text{fine}}) \in \mathbb{R}^{n_{\text{fine}} \times c \times 16}$$

where $|_v$ denotes the embedding of mesh vertex v, $\sum_p$ sums over the three (four) nearest coarse-scale points p in the case of a surface (volume) mesh, ϵ is a small constant and ψ is a geometric nonlinear layer. Note that this interpolation is a convex combination of multivectors. It thus admits interpolation of, e.g., points encoded in multivectors. This is expressed in the following proposition.

Proposition 3.1. *Consider a set of multivectors $x^i \in \mathbf{G}(3,0,1)$ and denote by*

$$t(x^i) \coloneqq \frac{1}{x^i_{123}} \begin{pmatrix} x^i_{012} \\ x^i_{013} \\ x^i_{023} \end{pmatrix}$$

the extraction of point coordinates from a multivector. Assume $x^i_{123} > 0$. Then the point extracted from convex combination of x^i is an element of the convex hull of $\{t(x^i)\}_i$.

3.4. Variants

The original LaB-GATr (Suk et al., 2024b) implements learned, geometric tokenisation by message passing: graph edges connect each (fine-scale) mesh vertex to the closest, sub-sampled (coarse-scale) point. Within these neighbourhoods, messages are computed via nonlinear geometric-algebra layers. LaB-GATr++ replaces this strictly local tokenisation module by the global cross-attention described above.

In a similar fashion, the interpolation module described above could be replaced by expanding cross-attention with the fine-scale features as queries. We investigate its effect on performance in an ablation study in Section 4.3.

4. Experiments

We evaluate LaB-GATr++ on the same cardiovascular hemodynamics regression tasks as in (Suk et al., 2024b). We train LaB-GATr++ under L^1 loss using the Adam optimiser (Kingma & Ba, 2015) with initial learning rate of 3e-4 and exponential decay. We use up to four NVIDIA L40 (48 GB) GPUs for parallelised training. With our hyperparameter setup, LaB-GATr++ has around 690k trainable parameters.

Table 1. **Accuracy** compared to the baselines on the wall shear stress (WSS) and velocity field datasets. We report mean approximation error ε across the test sets.

Dataset	Model	$\varepsilon[\%] \downarrow$
WSS	GEM-CNN (Suk et al., 2024a)	7.8
	GATr (Brehmer et al., 2023)	**5.5**
	LaB-GATr (Suk et al., 2024b)	**5.5**
	LaB-GATr++ (ours)	**5.5**
velocity	SEGNN (Suk et al., 2023)	7.4
	LaB-GATr (Suk et al., 2024b)	3.5
	LaB-GATr++ (ours)	**2.7**

4.1. Wall shear stress (WSS)

We compare LaB-GATr++ against several baselines in the task of 3D WSS estimation on surface meshes of synthetic coronary arteries (Suk et al., 2024a). The dataset consists of 2k differently sized and shaped meshes of around 7k vertices. We choose sampling ratio $\frac{n_{\text{coarse}}}{n_{\text{fine}}} = 10\%$ (see Section 3.2) and train LaB-GATr++ for 4k epochs (1 min 6 s per epoch) with batch size 8. Visual inspection of a test-split prediction (see Figure 2) shows excellent correspondence to the ground truth and the error is an order of magnitude smaller than the WSS. In Table 1 we report global approximation error ε (Suk et al., 2024a) and compare it to the baselines. LaB-GATr++ matches state-of-the-art.

4.2. Velocity field in bifurcating arteries

Furthermore, we evaluate LaB-GATr++ on the task of 3D velocity field estimation in volume meshes of bifurcating coronary arteries (Suk et al., 2023). The meshes in this dataset are considerably larger at around 175k vertices. To accommodate for this, we choose an aggressive sampling ratio of $\frac{n_{\text{coarse}}}{n_{\text{fine}}} = 1\%$ and train on four GPUs in parallel for 300 epochs (10 min 37 s per epoch) with batch size 1. Again, visual inspection of a test-split prediction (see Figure 2) shows excellent correspondence to the ground truth and the error is an order of magnitude smaller than the velocity. Table 1 shows that LaB-GATr++ improves over the previous state-of-the-art by 0.8 percentage points.

4.3. Ablation of variants

We study the influence of cross-attention for sequence reduction and expansion and compare it to the original pooling and interpolations modules of LaB-GATr. To this end, we create model variants with all four possible combinations: message passing & interpolation (Lab-GATr), cross-attention & interpolation (LaB-GATr++), cross-attention & cross-attention and message passing & cross-attention.

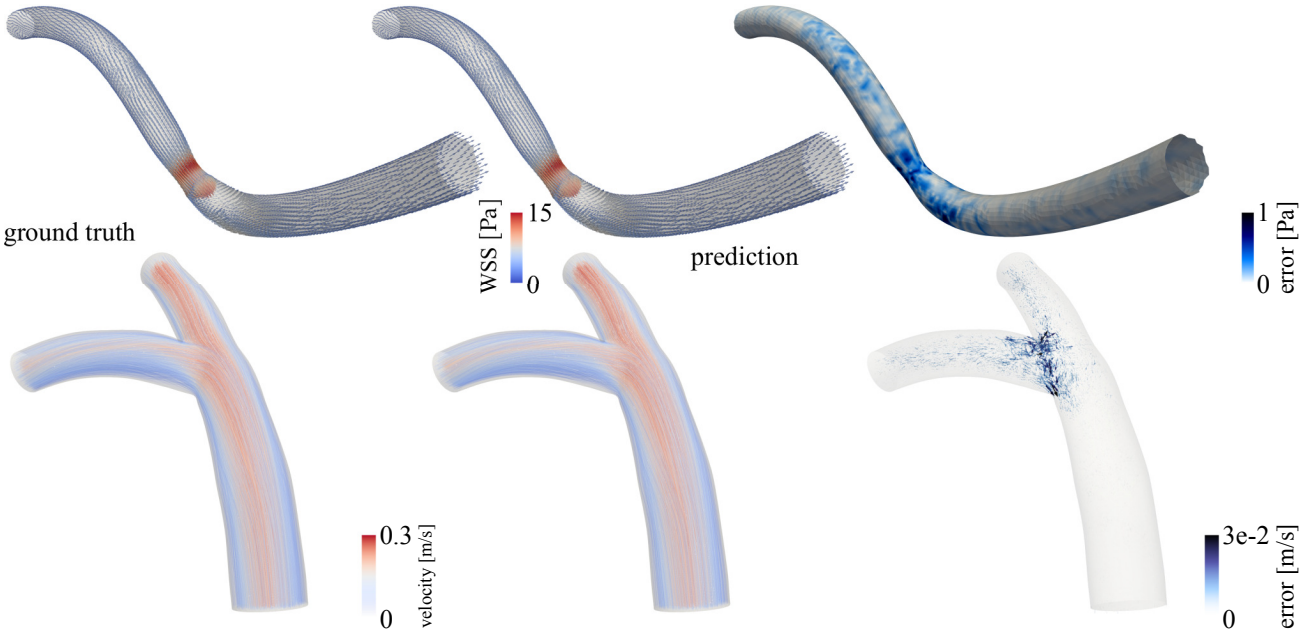

Figure 2. **Visual comparison** of ground truth obtained via fluid simulation and LaB-GATr++ prediction. We visualise 3D wall shear stress (WSS) vectors and error magnitude as surface map (top) as well as velocity streamlines and error vectors (bottom).

Table 2. **Ablation study** on the WSS dataset. We report mean approximation error ε across the test split, averaged over four training runs of 1k epochs.

Model (training time per epoch)	$\varepsilon[\%] \downarrow$
LaB-GATr (1 min 6 s)	6.35
LaB-GATr++ (1 min 6 s)	**6.23**
cross-attention & cross-attention (1 min 41 s)	6.50
message passing & cross-attention (1 min 33 s)	6.54

We train each model variant for 1k epochs on the WSS dataset in four separate training runs and average the evaluation errors. Results are shown in Table 2. We find that replacing interpolation (Section 3.3) by cross-attention in LaB-GATr++ (i.e. cross-attention & cross-attention) leads to inferior accuracy as well as incurs training time overhead of 53 %. In contrast, cross-attention seems to be favourable over message-passing tokenisation for both accuracy and runtime.

5. Discussion and conclusion

In this work, we propose an extension to LaB-GATr (Suk et al., 2024b) by replacing its message-passing pooling by cross attention. We demonstrate that this increases accuracy on a task in cardiovascular velocity field estimation. We attribute this to the global context of cross-attention compared to the strictly local context of message passing. Using a coarsening operation in geometric graph transformers is beneficial for two reasons: 1) it alleviates learning on extremely large meshes and 2) it retracts focus off the granularity and placement of mesh vertices, which – in continuum mechanics – are an artefact, not a feature, of the 3D geometry. While we experimented with cross attention as replacement of the interpolation module, we found inferior performance in terms of accuracy and runtime. The former might be specific to our data rather than the module itself. The latter is because the interpolation module (Section 3.3) is extremely light-weight. To gain more insight into both the coarsening and interpolation layer, it would be interesting to investigate how much features are shared across long distances and wether the difference in performance depends on size or granularity of the mesh and the level of coarsening.

With the presented primitives, we could realise a geometric version of sliding window (Swin) attention (Liu et al., 2021), which is an interesting direction for future work. However, we do not expect it to benefit applications with similarly structured data as the two presented ones. This is due to the long-range, global nature of blood flow in arteries and the meshes' oversampling of their 3D continuum. Geometric cross-attention could also be interesting for classification tasks involving 3D meshes as well multi-modal settings where geometric information should be fused with semantic infomration. Geometric algebra introduces an inductive

bias to our learning framework. In future work, we aim to investigate to which extent this affects hemodynamics estimation by deriving theoretical guarantees. Like GATr and LaB-GATr, LaB-GATr++ is equivariant under $\mathrm{E}(3)$ transformations, i.e. rotations, translations and reflections of the input geometry.

In conclusion, cross-attention is a powerful alternative to message passing for downsampling in LaB-GATr.

References

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021.

Arzani, A., Wang, J.-X., Sacks, M., and Shadden, S. Machine learning for cardiovascular biomechanics modeling: Challenges and beyond. Annals of Biomedical Engineering, 50:1–13, 04 2022.

Brandstetter, J., van den Berg, R., Welling, M., and Gupta, J. K. Clifford neural layers for PDE modeling. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, 2023.

Brehmer, J., de Haan, P., Behrends, S., and Cohen, T. Geometric algebra transformer. In Advances in Neural Information Processing Systems, volume 37, 2023.

Dahan, S., Fawaz, A., Suliman, M. A., da Silva, M., Williams, L. Z. J., Rueckert, D., and Robinson, E. C. The multiscale surface vision transformer. ArXiv, 2023.

Jaegle, A., Borgeaud, S., Alayrac, J., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Hénaff, O. J., Botvinick, M. M., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver IO: A general architecture for structured inputs & outputs. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

Li, G., Wang, H., Zhang, M., Tupin, S., Qiao, A., Liu, Y., Ohta, M., and Anzai, H. Prediction of 3d cardiovascular hemodynamics before and after coronary artery bypass surgery via deep learning. Communications Biology, 4, 01 2021.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV, Montreal, QC, Canada, October 10-17, 2021, 2021.

Rabe, M. N. and Staats, C. Self-attention does not need $O(n^2)$ memory. In n/a, 2021.

Ruhe, D., Brandstetter, J., and Forré, P. Clifford group equivariant neural networks. In Thirty-seventh Conference on Neural Information Processing Systems, 2023a.

Ruhe, D., Gupta, J. K., Keninck, S. D., Welling, M., and Brandstetter, J. Geometric clifford algebra networks. In International Conference on Machine Learning, ICML, 23-29 July 2023, Honolulu, Hawaii, USA, 2023b.

Sarasua, I., Pölsterl, S., and Wachinger, C. Transformesh: A transformer network for longitudinal modeling of anatomical meshes. In Machine Learning in Medical Imaging, pp. 209–218, Cham, 2021. Springer International Publishing. ISBN 978-3-030-87589-3.

Suk, J., Brune, C., and Wolterink, J. M. SE(3) symmetry lets graph neural networks learn arterial velocity estimation from small datasets. In Bernard, O., Clarysse, P., Duchateau, N., Ohayon, J., and Viallon, M. (eds.), Functional Imaging and Modeling of the Heart, pp. 445–454, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-35302-4.

Suk, J., de Haan, P., Lippe, P., Brune, C., and Wolterink, J. M. Mesh neural networks for SE(3)-equivariant hemodynamics estimation on the artery wall. Computers in Biology and Medicine, 173:108328, 2024a. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2024.108328. URL https://www.sciencedirect.com/science/article/pii/S0010482524004128.

Suk, J., Imre, B., and Wolterink, J. M. LaB-GATr: geometric algebra transformers for large biomedical surface and volume meshes. ArXiv, abs/2403.07536, 2024b. URL https://api.semanticscholar.org/CorpusID:268363685.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Annual Conference on Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA, 2017.

Wang, Y., Elhag, A. A. A., Jaitly, N., Susskind, J. M., and Bautista, M. A. Swallowing the bitter pill: Simplified scalable conformer generation. ArXiv, 2023.

A. Proof of proposition 3.1

Proof. Let $w = \sum_i \omega_i x^i$, such that $\omega_i > 0$ and $\sum_i \omega_i = 1$. Then

$$t(w) = \frac{1}{w_{123}} \begin{pmatrix} w_{012} \\ w_{013} \\ w_{023} \end{pmatrix} = \frac{1}{\sum_i \omega_i x^i_{123}} \begin{pmatrix} \sum_i \omega_i x^i_{012} \\ \sum_i \omega_i x^i_{013} \\ \sum_i \omega_i x^i_{023} \end{pmatrix}.$$

Define

$$\omega'_i = \frac{\omega_i x^i_{123}}{\sum_j \omega_j x^j_{123}}$$

and note that $\omega'_i > 0$ and $\sum_i \omega'_i = 1$. Now

$$\begin{aligned} \begin{pmatrix} \sum_i \frac{\omega_i}{\sum_j \omega_j x^j_{123}} x^i_{012} \\ \sum_i \frac{\omega_i}{\sum_j \omega_j x^j_{123}} x^i_{013} \\ \sum_i \frac{\omega_i}{\sum_j \omega_j x^j_{123}} x^i_{023} \end{pmatrix} &= \begin{pmatrix} \sum_i \frac{\omega'_i}{x^i_{123}} x^i_{012} \\ \sum_i \frac{\omega'_i}{x^i_{123}} x^i_{013} \\ \sum_i \frac{\omega'_i}{x^i_{123}} x^i_{023} \end{pmatrix} \\ &= \sum_i \omega'_i t(x^i) \end{aligned}$$

and thus

$$t(w) = t(\sum_i \omega_i x^i) = \sum_i \omega'_i t(x^i).$$

Since the convex hull of a set of points is defined as the set of all convex combinations of its elements, $t(\sum_i \omega_i x^i)$ is an element of the convex hull of $\{t(x^i)\}_i$. □