DOI: 10.1111/cgf.14912
Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 5

Attention And Positional Encoding Are (Almost) All You Need For Shape Matching

Alessandro Raganato and Gabriella Pasi and Simone Melzi

Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicoccca, Italy.

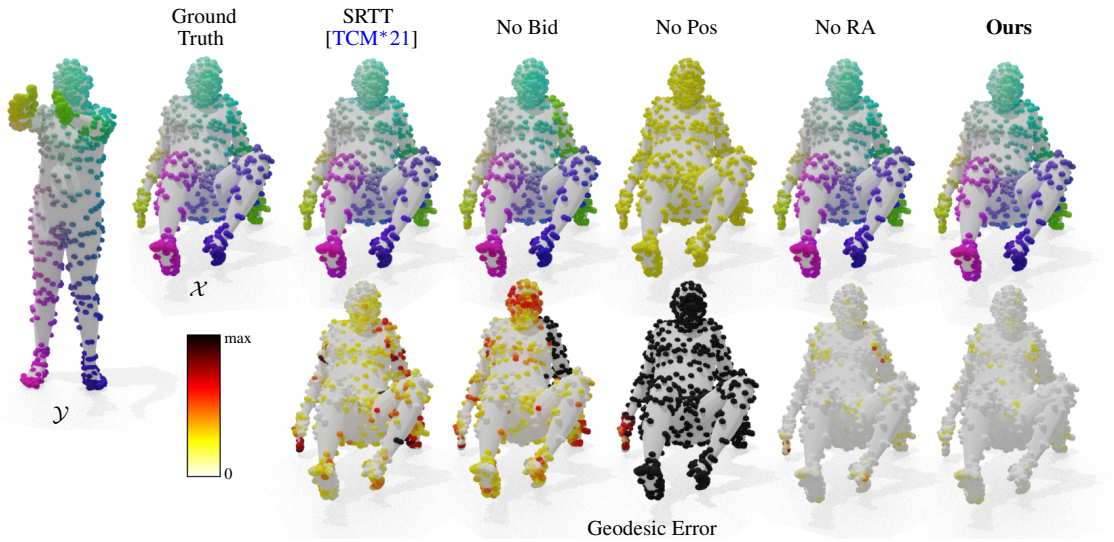

Figure 1: *Shape matching results produced for an example pair by some of the models we consider in the comprehensive ablation study that we perform. On the top row, we visualize correspondent points with the same color. On the bottom row, we encode the error of the estimated correspondence through the colorbar; exact matchings are white, while dark colors mean significant errors.*

Abstract

The fast development of novel approaches derived from the Transformers architecture has led to outstanding performance in different scenarios, from Natural Language Processing to Computer Vision. Recently, they achieved impressive results even in the challenging task of non-rigid shape matching. *However, little is known about the capability of the Transformer-encoder architecture for the shape matching task, and its performances still remained largely unexplored. In this paper, we step back and investigate the contribution made by the Transformer-encoder architecture compared to its more recent alternatives, focusing on why and how it works on this specific task. Thanks to the versatility of our implementation, we can harness the bi-directional structure of the correspondence problem, making it more interpretable. Furthermore, we prove that positional encodings are essential for processing unordered point clouds. Through a comprehensive set of experiments, we find that* attention and positional encoding are (almost) all you need *for shape matching. The simple Transformer-encoder architecture, coupled with relative position encoding in the attention mechanism, is able to obtain strong improvements, reaching the current state-of-the-art.*

CCS Concepts

• ***Computing methodologies*** → ***Shape analysis;*** • ***Theory of computation*** → *Computational geometry;*

1. Introduction

Over recent years the growing availability of acquisition devices and the corresponding significant increase in generated 3D data,

shape matching, and 3D registration have drawn interest in the scientific community. This attention is motivated by the numerous instances of these problems in different scenarios, from medical imaging to statistical shape analysis, from geological modeling to virtual and augmented reality, to name a few. In this work, we focus on the shape matching task, which aims to find a correspondence between the points that discretize a pair of shapes, a fundamental and initial step for 3D registration.

After their appearance on the scene of machine translation [VSP*17], Transformer architectures have been employed in several different applications of Natural Language Processing [DCLT19], Computer Vision [JGB*21], and Graphics [ZWC22] among many others. Taking root in the translation task, the Transformers naturally fit the 3D registration and non-rigid shape matching tasks that aim to "translate" the discrete representation of the geometry of one object into the discretization of a second one.

Inspired by this idea, Trappolini and colleagues [TCM*21] recently adopted the Perceiver Transformer [JGB*21], initially proposed for Computer Vision, as *geometrical translator* to solve the 3D registration problem. Key-aspect of their method is the definition of a novel attention, the *surface attention* that forces the model to take into account the width of the patch of the surface represented by each point in the discretization. This attention better encodes the continuous nature of the surface underlying the sampled 3D points giving rise to state-of-the-art performances and robustness to different sampling densities. This solution outperforms the competitors on different datasets laying the groundwork for a family of Transformer-based methods for the 3D registration task.

Beyond its accurate performance, a series of questions arise from this work. Is the *surface attention* sufficient or necessary to adapt the Transformers architecture for the shape matching task? Is the Perceiver [JGB*21] the best choice to target this specific task? What is the Transformer architecture learning to address the matching problem? What geometry of the shapes is the attention mechanism encoding?

With our preliminary analysis, we aim to answer the aforementioned questions to the purpose of revealing some insights on the fundamental role of each Transformer component. Specifically, we first explore different ways to assemble the components of the simple Transformer encoder architecture [VSP*17]. By this analysis, we aim to discover if Transformers can target the shape matching problem without resorting to any explicit geometric bias as done in [TCM*21]. The experimental evaluation confirms that our implementation outperforms the recent competitors in different scenarios without requiring additional data or computational costs. As depicted for a pair in Figure 1, we validate, through an extensive ablation, each component we include in our framework to clarify their job and discover their role in the whole procedure.

Our contribution is threefold: i) we show which implementation choices are better suited to target the shape-matching task with a simple Transformer encoder; ii) we analyze the role of positional encodings in shape matching; iii) we interpret the patterns that arise in the attention mechanism, highlighting their geometric structures, providing some insights about their functionality.

2. Related work

Transformers models: one in many. Transformers came on the stage in 2017, in the seminal paper [VSP*17]. This architecture was originally proposed for machine translation following the sequence-to-sequence paradigm. Later works successfully applied Transformer-based models to various Natural Language Processing tasks [ZHLL20, PSS20], typically adopting the pretrain-then-finetune paradigm [DCLT19]. Its flexibility has been then proven with success also in other domains, such as Computer Vision [DBK*21], Computer Graphics [LYZ22], Speech Processing [WML*20], Reinforcement Learning [CLR*21], and on mixed modalities, such as Vision and Language [KSK21]. At the same time, recent research has proposed several Transformer variants, from changing the order of layer components [PSL20], and incorporating persistent memory [SGL*19], to a more general-purpose architecture for handling long-context tasks [HJC*22]. Several Transformer variants have also been used for point clouds related tasks [ZWC22]. However, little is known about the ability of the plain Transformer encoder to address the shape matching task. To bridge this gap, we investigate how this architecture is capable of solving the aforementioned task and what properties arise from the training of this model.

Inside Transformers: many in one. Interpreting the representations of a Transformer-based model is an active area of research, fueled by the recent breakthroughs in different domains [RT18, VST21, NRK*21]. Several lines of research explored the architecture in depth from different angles, from formalizing and measuring what properties the models learn, and how the systems can be biased to build better representations [RKR20, RVCT21]. In line with these directions, one of the main analyzed components of the Transformer architecture is the so-called multi-head attention module [KRRR19, BMPH21]. A typical example of the outcome of interpreting the attention weights and connections is in the Natural Language Processing field, where attention scores seem to correlate to certain types of linguistic phenomena [TXC*19, CKLM19], revealing which one could be pruned to reduce parameters footprint without performance loss [VTM*19, MLN19]. In a similar fashion, recent work in computational biology has shown which structure emerges from the task of protein sequence modeling in Transformer encoder models [VMV*21]. However, to the best of our knowledge, no attempt has been made to target the shape matching problem with a plain Transformer encoder, analyzing which structure is uncovered from the training on this task.

Non-rigid shape matching. Given a pair of surfaces that undergo a non-rigid deformation, *non-rigid shape matching* consists of assigning each point from the first to the corresponding point on the second. Solving this problem is fundamental for many applications, such as statistical shape analysis, texture transfer, and shape interpolation, among many others. For this reason, numerous contributions on this topic have been raised in the last decades. This brief overview is not meant to be exhaustive, and we refer the interested reader to the available surveys [VKZHCO11, TCL*13, BCBB16, Sah20, DYDZ22].

A family of approaches is based on the iterative closest point procedure (ICP) [BM92, ARV07, LSP08], which iteratively alternate two steps: i) deform the 3D coordinates of the first sur-

face to fit the geometry of the second; ii) compute the correspondence as a nearest neighbor search in the 3D space. Descriptors-based methods are a valid alternative. They assign to each point a vector invariant to a specific set of deformations. Then, they compute correspondence by comparing these vectors and picking the most similar ones as corresponding points. Several descriptors [Rus07, SOG09, ASC11, MRCB16] arise from the Laplacian [PP93] and its eigendecomposition [Tau95, Lev06, LZ10] which promotes invariance to isometric deformations. Other signatures exploit a local analysis of the extrinsic geometry of the neighborhood of each point [TSDS10, MST*19]. Other methods target the shape-matching task by proposing different representations of the correspondences, such as parametric [APL15] or as blended across multiple maps [KLF11]. In [EEBC20], the proposed pipeline first computes an initial correspondence among landmarks by exploiting an evolutionary genetic algorithm. Then it extends this sparse mapping to a dense correspondence by minimizing a local metric distortion. Another valuable solution is the functional map framework [OBCS*12], which focuses on the functional mapping induced by the point-to-point correspondence. Several variants of this framework have been proposed, adding regularizers [OCB*17, NO17, NMR*18, RPWO18, DCMO22], addressing clutter and partialities [RCB*17, CRM*16], or exploiting the relation between pointwise and functional mapping [MRR*19, HRWO20, RMOW20, PRM*21, RMWO21, PKO22].

With the rise of machine learning, different data-driven strategies have been applied to boost the functional maps [LRR*17, DSO20, APO21]. In [MRMO20], the authors propose a learning procedure to apply functional maps to 3D point clouds exploiting the PointNet network [QSMG17]. 3DCoded [GFK*18] adapts an autoencoder architecture to register a fixed template to an input surface so that every pair of shapes can be matched through the registered template. Recently, Trappolini and colleagues [TCM*21] targeted the shape registration task by exploiting the power of the Transformer architecture from [JGB*21]. In their model, which from now on we name SRTT, they define *the surface attention*, which consists of a weighted attention mechanism that, for each point, takes into account the patch of the continuous surface that it represents. This information makes the model resilient to different surface samplings and densities. The obtained model outperforms all the competitors on different benchmarks. Both 3DCoded and SRTT solve the matching task as a registration problem, which also aims to fit the bi-dimensional surface that represents a first shape on the geometry of a second one. In this paper, we focus on shape-matching, which only targets discrete maps among the set of 3D points that represent the surface and thus is more general and can be seen as an initial step for registration.

3. Background and motivations

In this Section, we introduce the shape matching task emphasizing its main properties and challenges.

$\mathcal{Y}$ $\mathcal{X}$

The problem A discrete shape or surface $\mathcal{X}$, is a collection of $n_{\mathcal{X}} \in \mathbb{N}$ points $X \in \mathbb{R}^{n_{\mathcal{X}} \times 3}$ in the 3D space, which approximates a 2-dimensional smooth manifold embedded in $\mathbb{R}^3$. Given a pair (or a collection) of surfaces $\mathcal{X}$ and $\mathcal{Y}$ that undergo a non-rigid deformation T (*i.e.*, $\mathcal{Y} = T(\mathcal{X})$), and their discretizations X and Y, *non-rigid shape matching* consists of assigning each point of X the unknown corresponding point in Y and vice-versa. These points can own connectivity as a triangular or polygonal mesh, but we focus on the more general setting of point clouds without any additional structure. The inset figure shows an example of correspondence for a pair of surfaces $\mathcal{X}$, $\mathcal{Y}$, represented as point clouds, which discretize the underlying white surfaces that we render just for visualization purposes. We represent the correspondence through color-coding: the same color means correspondent points.

Unordered point clouds The lists of points that represent each surface do not share a common order, and, in the general case, their cardinality is different (*i.e.*, $n_{\mathcal{X}} \neq n_{\mathcal{Y}}$), usually belonging to the interval $[1K, 200K]$. The shape matching problem is thus a combinatorial problem, and there is no information in sorting the 3D points. In fact, given two random permutations of the rows of X and Y, a shape matching solution should be able to recover the same correspondence independently from the order of the points.

Rigid and non-rigid deformations Each shape/surface $\mathcal{X}$ may undergo to different transformations, making the task particularly challenging. These transformations can be broadly grouped into two main categories: *rigid*, such as scale, translation, and rotation, and *non-rigid*, such as pose variations, different subjects, and articulated movements. Moreover, to further increase the complexity of the task, each deformation can be a composition of both *rigid* and *non-rigid* transformation.

Different densities of the sampling Shape matching solutions tend to suffer from the different densities of the surface discretization. Even though the bijection nature of the task mitigates this problem, current systems still struggle to match two sets of points with different cardinality [MMR*19].

Bi-directional nature The shape matching problem is bidirectional in its nature, i.e., finding a solution for the correspondence from $\mathcal{X}$ to $\mathcal{Y}$, also provides some information about the matching in the opposite direction, from $\mathcal{Y}$ to $\mathcal{X}$. Such property can be leveraged to transfer knowledge in both directions. This provides a strong signal in case of a bijection between the two shapes, i.e., when they have the same cardinality, while less effective when the two shapes are represented by a different number of points. For example, if $\mathcal{X}$ has fewer points than $\mathcal{Y}$, the correspondence from $\mathcal{X}$ to $\mathcal{Y}$ leaves some points of $\mathcal{Y}$ unmatched.

4. Proposed implementation

The gist of our solution lies in the combination of the Transformer encoder [VSP*17], positional encoding, and attention specialization. In Figure 2, we report a schema of the implemented architecture.

Transformer encoder architecture The Transformer encoder architecture consists of a number of stacked multi-head attention and feed-forward blocks. The multi-head attention is a concatenation of several attention functions called heads, typically implemented with a scaled dot-product attention module [VSP*17].

More formally, given an input with sequence length n and dimensionality d, first, it is processed into three linear projections, queries $\mathbf{Q} \in \mathbb{R}^{n\times d}$, keys $\mathbf{K} \in \mathbb{R}^{n\times d}$ and values $\mathbf{V} \in \mathbb{R}^{n\times d}$, and then an attention energy ξ is computed over the queries and keys:

$$\xi = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \qquad \in \mathbb{R}^n \tag{1}$$

where d is the dimensionality of the key. Finally, this attention energy ξ is used to compute the weighted average of the values V:

$$\text{Att}(\xi, \mathbf{V}) = \xi\mathbf{V} \qquad \in \mathbb{R}^d \tag{2}$$

This mechanism is computed h times, defined as the number of attention heads. Each attention head is then concatenated and fed to a two-layer feed-forward block with ReLU activation. To stabilize training, multi-head attention and feed-forward blocks are interweaved by layer normalization modules.

Positional encoding (*RoPE*) There are various methods to integrate position information into a Transformer model, broadly divided into two groups, absolute and relative positional encoding [DSS22]. Moreover, there are two approaches for incorporating such positional information, either added together with the input to the model or by directly modifying the attention matrices in every layer. In our work, we used rotary position encoding (*RoPE*) to encode relative positions within the model [SLP*21]. This method applies rotation matrices, built from sine and cosine functions, to each query and key attention heads in every layer. The main intuition is that it gives the model the ability to have knowledge that reflects the relative distance between each input point. As we input to the Transformer encoder two shapes to be matched, the positional knowledge is essential to enhance the capability of the network to distinguish the two shapes. The *RoPE* encoding is incorporated as follows:

$$\xi = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{R}^d_\Theta\mathbf{K}^\top}{\sqrt{d}}\right) \qquad \in \mathbb{R}^n \tag{3}$$

where R^d_Θ is a block-diagonal matrix with rotation matrices on its diagonal for each input position. More specifically, following [SLP*21], we use $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1,2,...,d/2]\}$. Given an input position m, the matrix $R^d_{\Theta,m}$ has the following rotation matrices on its diagonal:

$$\begin{pmatrix} \cos m\theta_i & -\sin m\theta_i \\ \sin m\theta_i & \cos m\theta_i \end{pmatrix}$$

Residual Attention (*RA*) The multi-head attention mechanism is one of the core components of the Transformer model. Each attention head usually learns different patterns useful to address the underlying task. In our scenario, in order to match two different shapes, a model should learn at least two crucial patterns, the shape itself, so recognizing a single shape and its target one. We incorporate this hunch by introducing a residual attention mechanism to improve the model to specialize each attention head to specific patterns [HRKA21]. Residual attention (*RA*) simply adds a connection to each attention matrix, propagating attention scores of the previous layer. This improves the model to learn sparse and specialized attention heads, which is effective for the shape-matching task. Finally, given a i-th layer, the attention energy function is defined as:

$$\xi_i = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{R}_\Theta\mathbf{K}^\top}{\sqrt{d}} + \xi_{i-1}\right) \qquad \in \mathbb{R}^n \tag{4}$$

Matching computation Given two shapes $\mathcal{X}$ and $\mathcal{Y}$ with $n_\mathcal{X}$ and $n_\mathcal{Y}$ points, the input of our network is a matrix of size $(n_\mathcal{X} + 1 + n_\mathcal{Y}) \times 3$, representing the 3 dimensional coordinates of the two point clouds split by a separator SEP of size 1×3. Similarly, the output of the network is a matrix with the same size as the input, which we can split into two 3D point clouds $\widehat{X}$ of size $n_\mathcal{X}$ and $\widehat{Y}$ of size $n_\mathcal{Y}$ by removing the SEP. $\widehat{X}$ is composed by the same points of $\mathcal{X}$ but placed in the 3D space in order to fit the geometry of $\mathcal{Y}$. The same holds for $\widehat{Y}$. Following previous work [TCM*21], we can extract the point-to-point correspondence between $\mathcal{X}$ and $\mathcal{Y}$ solving for each point $x \in \mathcal{X}$ the nearest neighbor assignment problem in the 3D space among the point of $\widehat{Y}$, and similarly for all $y \in \mathcal{Y}$ and $\widehat{X}$. Thanks to the 1:1 correspondence among the shapes in the training set, we can train our model by minimizing the sum of the losses $\ell_{\mathcal{X},\mathcal{Y}}$ and $\ell_{\mathcal{Y},\mathcal{X}}$, defined as:

$$\ell = \ell_{\mathcal{X},\mathcal{Y}} + \ell_{\mathcal{Y},\mathcal{X}} = \|\widehat{Y} - \mathcal{X}\|_2^2 + \|\widehat{X} - \mathcal{Y}\|_2^2 \tag{5}$$

Additional information about the model architecture is given in the supplementary material. Furthermore, we release our complete implementation at: https://github.com/raganato/SGP23_AttPos4ShapeMatching.

5. Experimental settings and evaluation

Our work aims to spread light on the best practices for achieving shape matching with Transformers. We are thus not proposing a novel model, instead, starting from the simplest Transformer encoder, we investigate how to adapt it to tackle shape matching, encoding geometric features relevant to this desired task. Moreover, we analyze the role of each component we integrated into this architecture. In this spirit, the main competitor is SRTT from [TCM*21], the first, and based on a recent survey [ZWC22], the only Transformer architecture designed to solve this task. In our experiments, we include the datasets and the training and test settings from [TCM*21], inheriting the comparisons from its evaluation. In the following, we briefly list these datasets and competitors, referring to [TCM*21] for additional details.

5.1. Datasets and settings

Training data For the experiments on human shapes, we train on $10K$ triangular meshes with $1K$ vertices representing different subjects in different poses. These point clouds have been obtained by processing meshes (with 6890 vertices) from the SURREAL dataset [VRM*17]. These $1K$ vertices are the same for all the training shapes preserving the original 1:1 correspondence. This training set is the same from [MRMO20] and [TCM*21]. As for the competitors, our model only accesses the 3D coordinates of these points, ignoring the connectivity information. We additionally generate a second version of this training set by sampling $1K$ points of the same original meshes. We select the first 500 by applying the

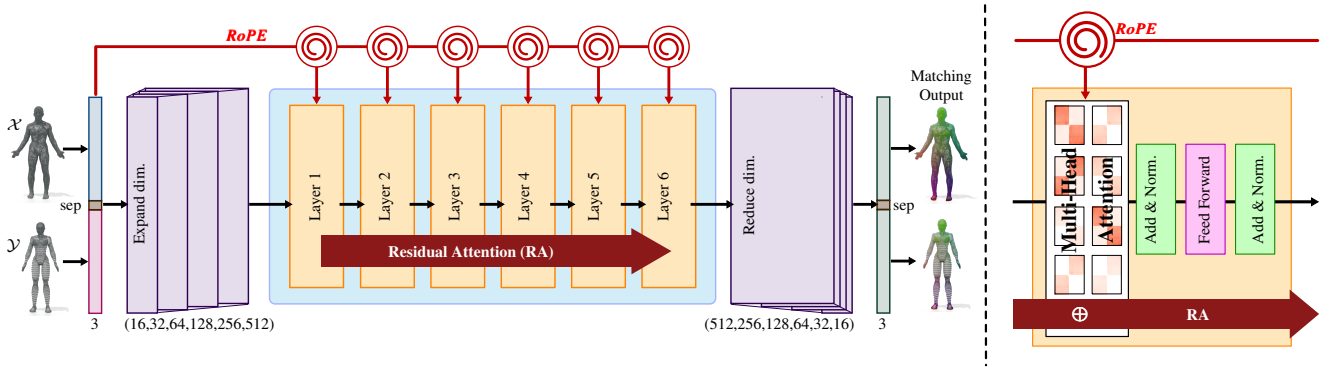

Figure 2: *A visualization of the overall architecture (on the left) and of a single Transformer layer (on the right). The model takes into input the concatenation of two shapes X and Y in $\mathbb{R}^{n\times 3}$, split by a separator SEP. After an initial series of dimensionality augmentation to fit the Transformer parameters, a stacked 6 layers Transformer-encoder block is applied to the concatenated input, coupled with rotary position encoding RoPE, and residual attention connection RA, placed in the multi-head attention component (right picture). Finally, a dimensionality reduction block returns the resulting target matching points in $\mathbb{R}^{n\times 3}$.*

euclidean farthest point sampling. Then we randomly pick the other 500 samples among the remaining ones. We denote this sampling strategy and all the experiments on these data with $\star$.

For the evaluation on animals, the training set consists of $20K$ shapes from [ZKJB17b], divided into five classes (cat, cow, dog, horse, hippo) and with various poses generated with the parametric model SMAL [ZKJB17a]. As done for the human shapes, by applying the strategy $\star$, we sample $1K$ points from the 3889 vertices generated by SMAL, preserving the 1:1 correspondence. Similarly, we select $20K$ shapes only from the hippos class to train a dedicated model. We respectively refer to these training sets as A and H.

Applied augmentations To enforce the robustness of the model to random rotation, we apply a *random rotation* which belongs, with probability $\frac{1}{3}$, to one of the following types: i) the composition of three random rotations, one for each axis in the interval $[0, 2\pi]$; ii) a random rotation along one of the axes in the interval $[0, 2\pi]$; iii) the null rotation. The desired output of the model should be independent of the permutations of the input points. To push this property, we apply a *random permutation* to each shape at train time too.

Test sets We consider the following standard datasets and their modified versions designed to assess the approaches in different settings. The first five are for humans shapes, from the FAUST dataset [BRLB14], while the last three are for animals. We know the ground truth correspondence for each pair in these sets for the evaluation. None of the test shapes were seen at training time.

$F_{\sim 7K}$ [BRLB14]: 10 subjects, in the same 10 poses, all represented with the same triangular mesh with 6890 vertices.

F_{1K}: The same shapes of $F_{\sim 7K}$, remeshed to the same triangular mesh with $1K$ vertices.

F_{1K}N: The same shapes of F_{1K} with Gaussian noise on the 3D coordinates with standard deviation equal to 0.01.

F_{1K}O: The same shape from $F_{\sim 7K}$ with a different sampling of $1K$ points. Some of them have been randomly moved far from the surface, becoming outliers.

S19 [MMR*19]: 44 shapes from different repositories, with various triangulation, numbers of points, poses, and subjects. A list of 430 pairs is provided to evaluate the resiliency of the method to different connectivities and densities.

SMAL: 100 random pairs selected among 300 shapes generated with SMAL (3889 vertices).

HIPPO: 100 random pairs of different hippos from SMAL.

TOSCA: [BBK08] 100 random pairs of synthetic triangular meshes belonging to different classes (dog, horse, wolf) with various poses ($\sim 10K$ vertices). We only consider pairs composed of shapes from the same class.

Out of training distribution of the test sets All the shapes involved in our experiments have never been seen during the training. We emphasize that the shapes in S19 and TOSCA differ from the training set, respectively on the human and animal classes. They represent different subjects and have completely different poses, often never seen at training time. Furthermore, the density of the points distributions in the original shapes is, in some cases, significantly different from the ones from which we generate the training data. The results on these test sets show the generalization ability of our method, at least in the intra-class scenario. Finally, the results on animals show that through an appropriate second training phase, our model can target a different class even if initially trained on human shapes.

Test on shapes with more than $1K$ points Due to the size of the training data and the design of the model we adopt, at test time, we can only input two point clouds of dimension $1K$. We note that the self-attention mechanism employed in the Transformer architecture has quadratic time and space complexity relative to the length of the

input. Although there are other memory-efficient variants that could fit longer input [TDBM22], investigating their implementation is beyond the scope of this work, and we leave this to future studies. This choice simplifies the architecture, but we should apply an alternative procedure for testing on shapes with more points, such as the ones from $F_{\sim 7K}$ and S19. Given the pair $\mathcal{X}, \mathcal{Y}$, we apply the $\star$ sampling to select $1K$ points from both shapes and compute the output on them. Then, maintaining the 500 determined by the Euclidean farthest point sampling algorithm on $\mathcal{Y}$, we iteratively pick at random 500 points among the ones we excluded in the previous iterations. For each iteration, we run the model, and we aggregate the output only for the new 500 random points to the previous one respecting the original order. When we finish all the points from $\mathcal{Y}$, we stop the process, eventually adding some samples twice in the last iteration to reach 500 points. By doing this, we obtain an aggregated $\widehat{\mathcal{Y}}$ with the exact dimension of $\mathcal{Y}$, and we can perform the matching comparing $\mathcal{X}$ and $\widehat{\mathcal{Y}}$. We refer the interested reader to the supplementary material for more details on this step.

Competitors In addition to SRTT, we consider all the other competitors from [TCM*21] that we introduced in Section 2. We namely refer to them as **3DC** [GFK*18], **DiffNet** [DSO20], **LinInv** [MRMO20] and to the results of the model we implemented as **Ours**. We also include a modification of **Ours**, injecting explicit geometrical information about the input shape, using the *surface attention* from [TCM*21] in the Transformer encoder architecture as we describe in the supplementary materials. We denote the results obtained exploiting this modification as **Ours**$_{SA}$.

Evaluation metrics Given the ground truth correspondence $\Pi^{GT}_{\mathcal{X},\mathcal{Y}}$ between $\mathcal{X}$ and $\mathcal{Y}$ (*i.e.*, $y = \Pi^{GT}_{\mathcal{X},\mathcal{Y}}(x) \in \mathcal{Y}$ is the correct match $\forall x \in \mathcal{X}$), we compute the geodesic error $\mathcal{E}^{\Pi}_{\mathcal{X},\mathcal{Y}}(x)$ of the estimated correspondence Π as:

$$\mathcal{E}^{\Pi}_{\mathcal{X},\mathcal{Y}}(x) = \mathcal{G}_{\mathcal{Y}}(\Pi^{GT}_{\mathcal{X},\mathcal{Y}}(x), \Pi(x)), \tag{6}$$

where $\mathcal{G}_{\mathcal{Y}}$ is the geodesic distance on the surface $\mathcal{Y}$. The average geodesic error, namely *AGE*, is the average value of this error on all the points that discretize $\mathcal{X}$. In the tables, for each dataset, we report the average value of the *AGE* on a collection of pairs. For visualization, we encode this error in a colormap, points with large errors have dark colors while 0 errors are white.

5.2. Results

Table 1 resumes the comparison for the five human datasets. Notably, we significantly improve the results on all the test sets derived from FAUST, reducing the error of more than 30% compared to SRTT$_R$, which is the refined version of the method from [TCM*21], which was the state-of-the-art on these datasets. In Figure 3, we visualize the error for the same pair from F_{1K}, F_{1K}N, $F_{\sim 7K}$. In all the cases, we reduce the error compared to SRTT.

On S19, even if competitive, we are slightly worse than SRTT. We charge this performance drop to the extreme sampling variation in the shapes from S19. We cannot perform better even with **Ours**$_{SA}$, which implements the surface attention. This result motivates our claim that the accuracy of SRTT could only partially arise from the *surface attention* proposed in [TCM*21]. Instead, we suppose that the registration could induce the performance gap by forcing the source surface to fit the target one in the 3D embedding and correcting some of the wrong assignments of the matching. Furthermore, we remark that only SRTT$_R$, which involves a refinement procedure, achieves the slightest error. Many postprocessing and refinement strategies to improve the quality of the output of our procedure are possible. For instance, we can convert the computed maps into functional maps and exploit the smooth prior provided by the functional representation. The relevance of the constraints provided by the registration task and the impact of the refinement pave the way for further explorations that do not depend on the choice of the transformer architecture and are out of the scope of this paper.

However, excluding S19, our model still generalizes to different sampling like or even better than SRTT. In Figure 4, we depict the estimated correspondences on a pair varying the discretization of the $\mathcal{Y}$ surface. From left to right, we consider F_1K with the same sampling of $\mathcal{X}$, the one of $F_{\sim}7K$, and the $\star$ sampling. This qualitative result shows that in some cases **Ours** is more robust than SRTT to sampling variations, proving that the *surface attention* is not sufficient to make the model invariant to different sampling. For this reason and to exploit the Transformer architecture power without any explicit geometric bias, we prefer to avoid surface attention.

To enforce resilience on these challenges, we continue the training of SRTT , **Ours** and **Ours**$_{SA}$ for 24 hours on the $\star$ training set. We report the results after this second learning phase in the last three rows of Table 1. **Ours**$_\star$ performs better than **Ours** in the most challenging test settings (the rightmost three) where the sampling density is most different from the standard training set. Similarly for **Ours**$_{SA\star}$ and **Ours**$_{SA}$ while SRTT$_\star$ does not improve through this additional training. We compare **Ours** and SRTT and their version $\star$ on pairs where $\mathcal{X}$ is from F_{1K} and $\mathcal{Y}$ is from $F_{\sim 7K}$. We refer to this setting as F_{1to7K}. A qualitative test on a pair from F_{1to7K} is depicted in the center of Figure 4, while we report the quantitative results in the supplementary materials. All the models provide similar performance when testing the same pairs from F_{1to7K} and $F_{\sim 7K}$. **Ours**$_\star$ is more stable than **Ours**, proving that the second training phase helps to target pairs with different samplings while it does not help for SRTT. We remark that, working with a simple Transformer encoder, training with a different sampling provides a more significant improvement than adding the surface attention to the model. Indeed, **Ours**$_\star$ is the best result among the methods without refinement. These findings indicate that the training data sampling process plays a crucial role in enhancing the model's robustness against sampling variations, prompting a compelling research question regarding potential approaches to mitigate such variations.

In Figure 5, we report a pair of examples for the texture transfer application. On the left is a pair from S19, on the right from F_1K. For both pairs, we report the ground truth for comparison. Even if numerically worse than SRTT on S19, we are accurate enough to target this application.

In Table 2, we report the results for the animal test set, one for each column. For SRTT, **Ours** and **Ours**$_{SA}$, we consider three training settings. Starting from the respective model trained on human shape, for the models denoted with A and H, we continue the training for 24 hours on the corresponding training set. The last training setting starts from the A model and executes a one-shot

Table 1: *Comparison to existing methods*

Method	F_{1K}	$F_{1K}N$	$F_{1K}O$	$F_{\sim 7K}$	S19
3DC	0.0542	0.0712	0.2306	0.0776	0.2138
DiffNet	0.0534	0.0985	0.3509	0.0656	0.1509
LinInv	0.0471	0.0618	0.1738	0.0942	0.1284
SRTT	0.0419	0.0510	0.1657	0.0513	0.0802
Ours	**0.0135**	**0.0286**	**0.0518**	0.0236	0.0930
Ours$_{SA}$	0.0146	0.0302	0.0520	**0.0229**	0.0981
3DC$_R$	0.0367	0.0526	0.2101	0.0485	0.1935
SRTT$_R$	0.0263	0.0410	0.1479	0.0369	**0.0615**
SRTT$_\star$	0.0364	0.0477	0.0952	0.0436	0.0971
Ours$_\star$	**0.0133**	**0.0279**	**0.0224**	**0.0199**	0.0773
Ours$_{SA\star}$	0.0170	0.0320	0.0558	0.0217	0.0890

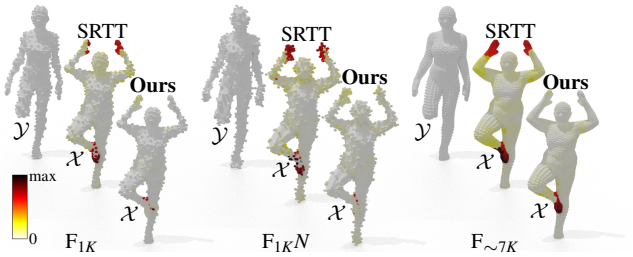

Figure 3: *A visualization of the geodesic error $\mathcal{E}^{\Pi}_{\mathcal{X},\mathcal{Y}}$ for SRTT [TCM*21] (top) and **Ours** (bottom) on the same pair from different test sets (from left to right: F_{1K}, $F_{1K}N$, $F_{\sim 7K}$).*

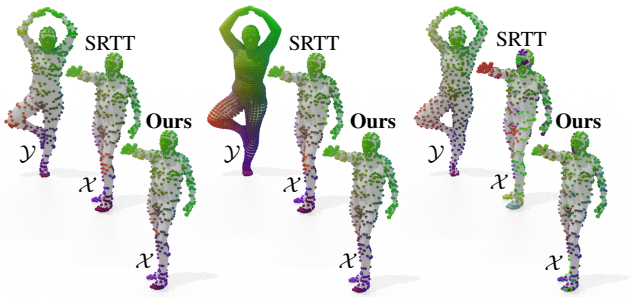

Figure 4: *A qualitative comparison between SRTT [TCM*21] (top) and **Ours** (bottom) on the same pair varying the sampling adopted to discretize $\mathcal{Y}$. From left to right: the same F_{1K} used for $\mathcal{X}$, 7K points from $F_{\sim 7K}$, F_{1K} distributed as in the training set $\star$.*

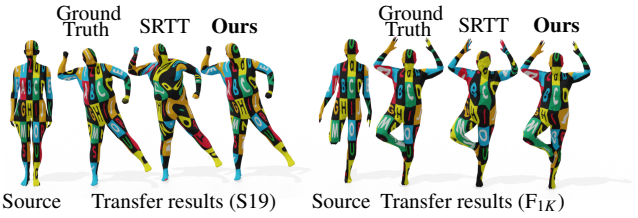

Figure 5: *Texture transfer results for two pairs (on the left from S19, on the right from F_{1K}). For each pair, from left to right, we visualize the source shape with the texture, the ground truth transfer, the output of SRTT, and our result.*

Table 2: *Comparison to SRTT [TCM*21] on the animal shapes.*

Method	SMAL	HIPPO	TOSCA
SRTT$_A$	0.0683	0.0369	0.3293
Ours$_A$	**0.0505**	0.0279	0.1499
Ours$_{SAA}$	0.0687	0.0422	0.1539
SRTT$_H$	0.2599	0,0303	0.3549
Ours$_H$	0.1623	**0.0173**	0.2722
Ours$_{SAH}$	0.1693	0.0218	0.3148
SRTT$_C$	0.0820	0.0516	0.3172
Ours$_C$	0.0659	0.0385	**0.0997**
Ours$_{SAC}$	0.0831	0.0579	0.1052

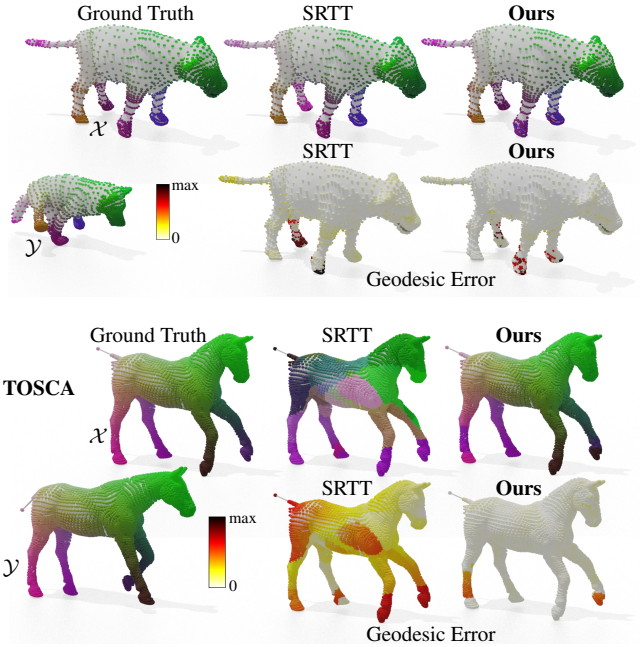

Figure 6: *Comparison between **Ours** and SRTT [TCM*21] on SMAL (upper images) and TOSCA (lower images).*

training on a single pair of cat shapes from TOSCA. We denote this training with *C*. For this reason, we exclude the cat class from the tests. In Figure 6, we visualize the comparison for two pairs, one from SMAL (top) and one for the horse class from TOSCA (bottom), respectively, with the *A* and *C* models. On SMAL, even if the errors are similar, our wrong matches are sparser than the ones from SRTT, which fails on an entire cow's leg and the tail. On TOSCA, we outperform SRTT. These results and Table 2 prove that **Ours** can better and faster learn to generalize to other categories both respect SRTTand **Ours**$_{SA}$. We remark that also in these scenarios, the surface attention does not generate any improvement.

6. Analysis and Ablation

Even if simple (*e.g.*, compared to the Perceiver [JGB*21]), our architecture provides excellent performance on different datasets dealing with several challenges. Our results are even more exciting since unlike SRTT and its *surface attention*, we do not inject explicit biases in the network to process geometric data or target the

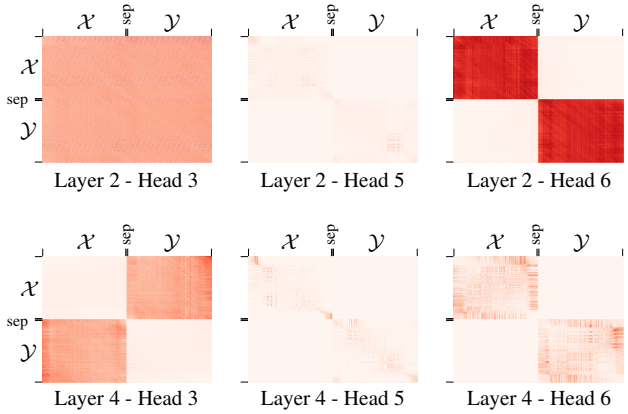

Figure 7: *A visualization of some attention heads from* ***Ours*** *model.*

desired task. Thanks to its design, our model infers all its capacity from the data. In the following, we analyze the main components of our architecture, revealing how they work and clarifying their role in achieving these impressive outcomes.

Analysis of the attention By visualizing the pattern of the attention heads across the layers, we reveal some exciting insights about our model. In Figure 7, we depict three attention heads in Layers 2 and 4. The stronger the red color, the higher the value of the attention. The blocks on the main diagonal represent self-attention, while the ones on the other diagonal correspond to cross-attention. The model, through the layers, specializes in each head, either self or cross-attention. Due to lack of space, we move to the supplementary material, the figure with all the attention heads for our model and for other models from the following ablation study discovering remarkable structures.

Geometry in the attention To further inspect the information encoded by the attention, in Figure 8, we plot some of the rows of a self-attention head (top) and a cross-attention head (bottom) as functions on the respective surfaces $\mathcal{X}$ and $\mathcal{Y}$. We only depict the values encoding information, the blocks on the main diagonal for the self and the ones from the other diagonal for the cross-attention (highlighted by the black dotted boxes). We adopt the same colormap used for the attention head: white is 0, and higher values are in darker red. With different colors we highlight the selected row in the attention head with an arrow (left), the corresponding point on the surfaces for the self-attention with a circle(top row), and the respective cross-attention again with an arrow (bottom row). The values of the self-attention are higher and resemble a Gaussian centered in the selected point. For every point, the corresponding cross-attention has smaller values and is less concentrated but represents a region that could correspond to the chosen point more smoothly. While the self-attention behaves similarly for different points, the cross-attention seems more sensitive. These results confirm that the learned attention heads do not only specialize in self or cross-attention but also encode the local region around each point. With this local information, the model learns about the local geometry around the point and exploits these additional features. We note that these findings belong to the **Ours** model, which does not require any geometric information about the input shapes. Moreover, our analysis discloses some intriguing questions. Is the learned attention dependent on the deformations the region around the selected point may undergo? As done in [YSI20, RST20], can we fix the attention pattern as Gaussian centered around the selected point for the self-attention heads and around the corresponding point for the cross-attention in the Transformer encoder? These are research directions with great potential that we reserve for future work.

Table 3: *Ablation study*

Method	♯Params	F_{1K}	$F_{1K}N$	$F_{\sim 7K}$	DEV
Ours	19.2M	0.0880	0.0826	0.0847	0.0489
Ours$_{SA}$	19.2M	0.0619	0.0683	0.0603	0.0432
XS-Ours	1M	0.1046	0.1129	0.1039	0.1039
4 Layers	12.9M	0.1749	0.1675	0.1425	0.0889
8 layers	25.5M	0.0856	0.0884	0.0749	0.0533
Pos [VSP*17]	20.2M	0.4238	0.3986	0.3476	0.4214
No Pos	19.2M	0.4155	0.3996	0.3466	0.4213
Only *RoPE*	19.2M	0.4155	0.3997	0.3466	0.4213
No Bid	19.2M	0.1284	0.1896	0.0733	0.0976
No RA	19.2M	0.0908	0.0935	0.0960	0.0573
SRTT [TCM*21]	1.7M	0.1678	0.1763	0.1728	0.0929
SRTT$_{XL}$	18.4M	0.1220	0.1332	0.1528	0.0723

Ablation study We perform a comprehensive ablation analysis training for 24 hours on a reduced version of $1K$ point clouds from the original train set for human shapes. In the same setting, for comparison, we train two versions of the SRTTmodel: i) the one from [TCM*21]; ii) SRTT$_{XL}$ for which we augment the dimensionality of the overall architecture to obtain a model with more parameters as in **Ours**. We evaluate the models on F_{1K}, $F_{1K}N$, $F_{\sim 7K}$, and DEV, a collection of shapes not included in the training set but generated in the same way. Each row in Table 3 corresponds to a model we train by disabling or modifying a specific component of the proposed architecture. In the second column, we report the number of parameters. In Figure 1, we depict a qualitative example produced by some of these models on a pair from DEV. **Ours** and **Ours**$_{SA}$ refer to the same architectures we test in the previous section. In **XS-Ours**, we reduce the dimensionality of the Transformer's input from 512 to 64. This modification streamlines the model's parameters but gives rise to a drop in the performance. With **4 Layers** and **8 Layers**, we compare Transformers composed of different numbers of layers. The poor results of **4 Layers** and the slight improvement provided by **8 Layers** at the cost of 25% of additional parameters motivate our choice of 6 Layers.

Surface attention In this setting, we note that **Ours**$_{SA}$ performs better than **Ours**. This improvement is not confirmed in the previous evaluations with more extended training and considerably more training data. The surface attention acts as a geometric prior that helps to learn faster even with reduced data but does not impact the performance in the full training.

Importance of the positional encodings To assess the role of the positional information and the preferable way to inject it into the architecture, we run three different models. **Only** *RoPE* receives as input the positional encoding without the features extracted from the 3D coordinates. This input is not sufficient to solve

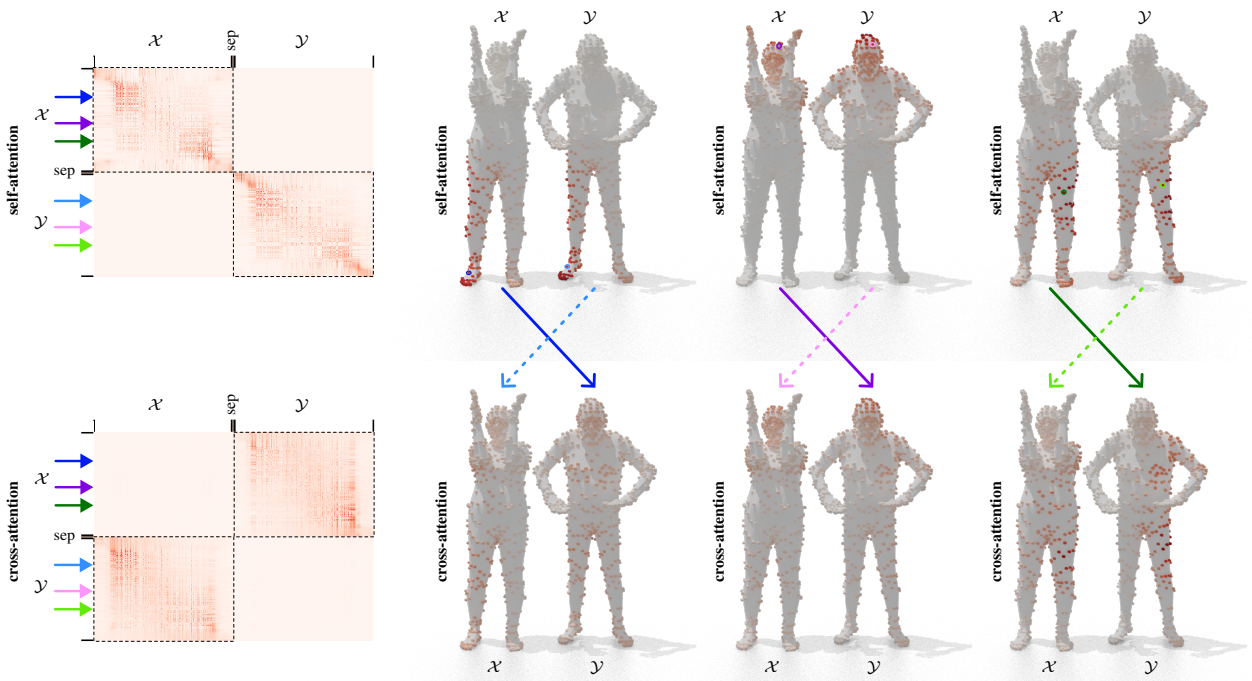

Figure 8: *A visualization of the self (top row) and cross-attention (bottom row) for three different points for each shape of a pair from the FAUST dataset. We report the attention head on the left and highlight the selected points with a colored arrow. On the right, for each selected row, we plot as a function on the surfaces the corresponding attention values restricted to the self-attention blocks in the first row and the cross-attention block in the second row (highlighted with the dotted black boxes). In the first row, we underline the surface point corresponding to the selected row in the attention matrix with a circle of the same color. With an arrow of the same color, we connect each chosen point on a shape with the value of the cross-attention it generates on the second. We adopt the same colormap for the attention matrix and for the rendering on the point clouds: attention value increases from light to darker colors.*

the task proving that positional information is important but not sufficient for shape matching. In **Pos** [VSP*17], we apply the absolute positional encoding proposed in [VSP*17]. This information is given to the network once at the beginning with the input. The bad results of **Pos** indicate that this method is inadequate in addressing the task effectively. Similarly, **No Pos**, which lacks the integration of any positional information, fails to solve the task too. We hypothesize that the model is not able to recognize the twofold structure of the input, and randomly embed all the points in a common space as suggested by the visualization on the right of Figure 9. Finally, although there are several other approaches available for integrating positional information to recognize the input shapes [DSS22], conducting a comprehensive search is beyond the scope of this study, and we defer it to future investigations.

Bi-directional nature of the problem We train the **No Bid** model minimizing the loss $\ell_{\mathcal{X},\mathcal{Y}}$ alone. As might be expected, this model generates larger errors on pairs for which the correspondence is a bijection. Moreover, in the supplementary material, we show that **No Bid** produces less cross-attention patterns.

Importance of recurrent attention [HRKA21] As we saw, our model specializes the attention-heads across the training. We appreciate that this property is less evident when we disable the recurrent attention in the **No RA** model giving rise to less accurate performances.

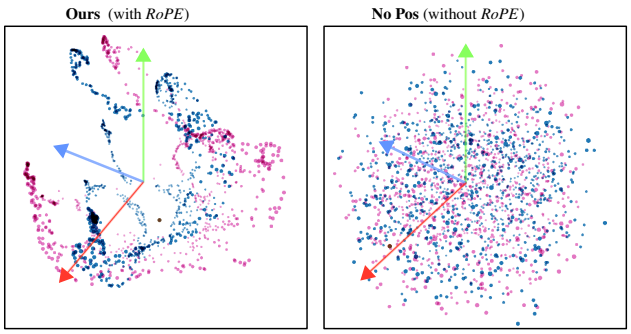

Figure 9: *A PCA visualization of the embedding produced by our model as the output of the 6^{th} layer, with (left) and without (right) the positional encoder RoPE [SLP*21].*

Role of the augmentations In Table 4, we test the robustness of our model to random permutations and random rotations of the input point clouds. From the augmentations applied at training time, **Ours** inherits *rotation* and *permutation invariance*. SRTT is ro-

Table 4: *Comparison to SRTT [TCM*21] applying random rotations on the 3-axes in the interval $[0, 2\pi]$, or random permutations to the point lists of the shapes.*

	Method	F_{1K}	F_{1K} N	F_{1K} O	$F_{\sim 7K}$	S19
	SRTT	0.0419	0.0510	0.1657	0.0513	0.0802
	Ours	0.0135	0.0286	0.0518	0.0236	0.0930
	Ours$_{SA}$	0.0146	0.0302	0.0520	0.0229	0.0981
Rand. rotat.	SRTT	0.2441	0.2674	0.3160	0.2629	0.2841
	Ours	0.0150	0.0292	0.0520	0.0239	0.0954
	Ours$_{SA}$	0.0161	0.0311	0.0531	0.0235	0.0999
Rand. perm.	SRTT	0.0417	0.0511	0.1621	0.0513	0.0802
	Ours	0.0126	0.0276	0.0503	0.0242	0.0943
	Ours$_{SA}$	0.0142	0.0300	0.0519	0.0232	0.0992

bust to permutations but can not deal with random rotations due to the limited set included in its augmentation.

7. Conclusions

By proposing a simple Transformer architecture to target the shape-matching application with accurate precision, we offer, for the first time, a exploratory analysis of the role and importance of many components to train a Transformer architecture to target the desired goal. Our study reveals some meaningful insights and visualizations into the injection of positional information and the patterns generated by the attention mechanism. The model we train achieves state-of-the-art performances on different datasets; when not, it is always comparable to existing alternatives. The limitations of our method are twofold: i) it can only be applied to pair with the same cardinality; ii) it suffers severe differences in the sampling density and distribution on the two shapes. Further explorations are necessary to solve these issues, but preliminary results support the adoption of additional data with different sampling during training as a potential solution. Moreover, as future work, we will target other challenging settings carrying out similar analyses for partial shape matching where one or both the shapes have missing parts or holes or consider shapes with topological noise induced by the acquisition process. Finally, our analysis reveals that our model generates attention heads similar to diagonal blocks in both directions, and as an exciting future direction, we can directly provide these patterns to the network to speed up the learning procedure and improve performance [RST20, YSI20, RVCT21].

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the RTX A5000 GPUs granted through the Academic Hardware Grant Program to the the University of Milano-Bicocca for the project "Learned representations for implicit binary operations on real-world 2D-3D data", and the CSC – IT Center for Science, Finland, for computational resources.

References

[APL15] AIGERMAN N., PORANNE R., LIPMAN Y.: Seamless surface mappings. *ACM Trans. Graph. 34*, 4 (jul 2015). 3

[APO21] ATTAIKI S., PAI G., OVSJANIKOV M.: Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)* (Los Alamitos, CA, USA, 2021), IEEE Computer Society, pp. 175–185. 3

[ARV07] AMBERG B., ROMDHANI S., VETTER T.: Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE conference on computer vision and pattern recognition* (2007), IEEE, pp. 1–8. 2

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 1626–1633. 3

[BBK08] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY, 2008. 5

[BCBB16] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.: Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum 35*, 6 (2016), 87–119. 2

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (Feb 1992), 239–256. 2

[BMPH21] BHARDWAJ R., MAJUMDER N., PORIA S., HOVY E.: More identifiable yet equally performant transformers for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, Aug. 2021), Association for Computational Linguistics, pp. 1172–1182. 2

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE. 5

[CKLM19] CLARK K., KHANDELWAL U., LEVY O., MANNING C. D.: What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (Florence, Italy, Aug. 2019), Association for Computational Linguistics, pp. 276–286. 2

[CLR*21] CHEN L., LU K., RAJESWARAN A., LEE K., GROVER A., LASKIN M., ABBEEL P., SRINIVAS A., MORDATCH I.: Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems* (2021), Beygelzimer A., Dauphin Y., Liang P., Vaughan J. W., (Eds.). 2

[CRM*16] COSMO L., RODOLÀ E., MASCI J., TORSELLO A., BRONSTEIN M. M.: Matching deformable objects in clutter. In *International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 1–10. 3

[DBK*21] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBORN D., ZHAI X., UNTERTHINER T., DEHGHANI M., MINDERER M., HEIGOLD G., GELLY S., USZKOREIT J., HOULSBY N.: An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021), OpenReview.net. 2

[DCLT19] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 4171–4186. 2

[DCMO22] DONATI N., CORMAN E., MELZI S., OVSJANIKOV M.: Complex functional maps: A conformal link between tangent bundles. *Computer Graphics Forum 41*, 1 (2022), 317–334. 3

[DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020). 3, 6

[DSS22] DUFTER P., SCHMITT M., SCHÜTZE H.: Position Information

in Transformers: An Overview. *Computational Linguistics 48*, 3 (09 2022), 733–763. 4, 9

[DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.: A survey of non-rigid 3d registration. *Computer Graphics Forum 41*, 2 (2022), 559–589. 2

[EEBC20] EDELSTEIN M., EZUZ D., BEN-CHEN M.: Enigma: Evolutionary non-isometric geometry matching. *ACM Trans. Graph. 39*, 4 (aug 2020). 3

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 230–246. 3, 6

[HJC*22] HAWTHORNE C., JAEGLE A., CANGEA C., BORGEAUD S., NASH C., MALINOWSKI M., DIELEMAN S., VINYALS O., BOTVINICK M., SIMON I., SHEAHAN H., ZEGHIDOUR N., ALAYRAC J.-B., CARREIRA J., ENGEL J.: General-purpose, long-context autoregressive modeling with perceiver AR. In *Proceedings of the 39th International Conference on Machine Learning* (17–23 Jul 2022), Chaudhuri K., Jegelka S., Song L., Szepesvari C., Niu G., Sabato S., (Eds.), vol. 162 of *Proceedings of Machine Learning Research*, PMLR, pp. 8535–8558. 2

[HRKA21] HE R., RAVULA A., KANAGAL B., AINSLIE J.: RealFormer: Transformer likes residual attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (Online, Aug. 2021), Association for Computational Linguistics, pp. 929–943. 4, 9

[HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum 39*, 5 (2020), 265–278. 3

[JGB*21] JAEGLE A., GIMENO F., BROCK A., VINYALS O., ZISSERMAN A., CARREIRA J.: Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), Meila M., Zhang T., (Eds.), vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 4651–4664. 2, 3, 7

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 79. 3

[KRRR19] KOVALEVA O., ROMANOV A., ROGERS A., RUMSHISKY A.: Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 4365–4374. 2

[KSK21] KIM W., SON B., KIM I.: Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), Meila M., Zhang T., (Eds.), vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 5583–5594. 2

[Lev06] LEVY B.: Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)* (2006), IEEE, pp. 13–13. 3

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 5659–5667. 3

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. *Computer graphics forum 27*, 5 (2008), 1421–1430. 2

[LYZ22] LI X.-J., YANG J., ZHANG F.-L.: Laplacian mesh transformer: Dual attention and topology aware network for 3d mesh classification and segmentation. In *Computer Vision – ECCV 2022* (Cham, 2022), Avidan S., Brostow G., Cissé M., Farinella G. M., Hassner T., (Eds.), Springer Nature Switzerland, pp. 541–560. 2

[LZ10] LEVY B., ZHANG R. H.: Spectral geometry processing. In *ACM SIGGRAPH Course Notes* (2010). 3

[MLN19] MICHEL P., LEVY O., NEUBIG G.: Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems* (2019), Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.), vol. 32, Curran Associates, Inc. 2

[MMR*19] MELZI S., MARIN R., RODOLÀ E., CASTELLANI U., REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval* (2019), Biasotti S., Lavoué G., Veltkamp R., (Eds.), The Eurographics Association. 3, 5

[MRCB16] MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M.: Shape analysis with anisotropic windowed fourier transform. In *International Conference on 3D Vision (3DV)* (2016). 3

[MRMO20] MARIN R., RAKOTOSAONA M.-J., MELZI S., OVSJANIKOV M.: Correspondence learning via linearly-invariant embedding. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 1608–1620. 3, 4, 6

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 155. 3

[MST*19] MELZI S., SPEZIALETTI R., TOMBARI F., BRONSTEIN M. M., STEFANO L. D., RODOLÀ E.: Gframes: Gradient-based local reference frame for 3d shape matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), IEEE, pp. 4629–4638. 3

[NMR*18] NOGNENG D., MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. *Computer Graphics Forum 37*, 2 (2018), 179–190. 3

[NO17] NOGNENG D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum 36*, 2 (2017), 259–267. 3

[NRK*21] NASEER M., RANASINGHE K., KHAN S., HAYAT M., KHAN F., YANG M.-H.: Intriguing properties of vision transformers. In *Advances in Neural Information Processing Systems* (2021), Beygelzimer A., Dauphin Y., Liang P., Vaughan J. W., (Eds.). 2

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 30:1–30:11. 3

[OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. In *SIGGRAPH 2017 Courses*. 2017. 3

[PKO22] PANINE M., KIRGO M., OVSJANIKOV M.: Non-isometric shape matching via functional maps on landmark-adapted bases. *Computer Graphics Forum 41*, 6 (2022), 394–417. 3

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental mathematics 2*, 1 (1993), 15–36. 3

[PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 384–393. 3

[PSL20] PRESS O., SMITH N. A., LEVY O.: Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 2996–3005. 2

[PSS20] PASINI T., SCOZZAFAVA F., SCARLINI B.: CluBERT: A cluster-based approach for learning sense distributions in multiple languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 4008–4018. 2

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660. 3

[RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M. M., TORSELLO A., CREMERS D.: Partial functional correspondence. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 222–236. 3

[RKR20] ROGERS A., KOVALEVA O., RUMSHISKY A.: A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics 8* (2020), 842–866. 2

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph. 39*, 6 (nov 2020). 3

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. *Computer Graphics Forum 40*, 5 (2021), 81–96. 3

[RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–16. 3

[RST20] RAGANATO A., SCHERRER Y., TIEDEMANN J.: Fixed encoder self-attention patterns in transformer-based machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), Association for Computational Linguistics, pp. 556–568. 8, 10

[RT18] RAGANATO A., TIEDEMANN J.: An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (Brussels, Belgium, Nov. 2018), Association for Computational Linguistics, pp. 287–297. 2

[Rus07] RUSTAMOV R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing* (2007), Eurographics Association, pp. 225–233. 3

[RVCT21] RAGANATO A., VÁZQUEZ R., CREUTZ M., TIEDEMANN J.: An empirical investigation of word alignment supervision for zero-shot multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Online and Punta Cana, Dominican Republic, Nov. 2021), Association for Computational Linguistics, pp. 8449–8456. 2, 10

[Sah20] SAHILLIOĞLU Y.: Recent advances in shape correspondence. *The Visual Computer 36*, 8 (2020), 1705–1721. 2

[SGL*19] SUKHBAATAR S., GRAVE E., LAMPLE G., JEGOU H., JOULIN A.: Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470* (2019). 2

[SLP*21] SU J., LU Y., PAN S., WEN B., LIU Y.: Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864* (2021). 4, 9

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum 28*, 5 (2009), 1383–1392. 3

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH, Association for Computing Machinery, p. 351–358. 3

[TCL*13] TAM G., CHENG Z.-Q., LAI Y.-K., LANGBEIN F., LIU Y., MARSHALL D., MARTIN R., SUN X., ROSIN P.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics 19* (7 2013), 1199–217. 2

[TCM*21] TRAPPOLINI G., COSMO L., MOSCHELLA L., MARIN R., MELZI S., RODOLÀ E.: Shape registration in the time of transformers. In *NeurIPS* (2021), Ranzato M., Beygelzimer A., Dauphin Y., Liang P., Vaughan J. W., (Eds.), pp. 5731–5744. 1, 2, 3, 4, 6, 7, 8, 10

[TDBM22] TAY Y., DEHGHANI M., BAHRI D., METZLER D.: Efficient transformers: A survey. *ACM Comput. Surv. 55*, 6 (dec 2022). 6

[TSDS10] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *Proc. ECCV* (2010), Springer, pp. 356–369. 3

[TXC*19] TENNEY I., XIA P., CHEN B., WANG A., POLIAK A., MCCOY R. T., KIM N., DURME B. V., BOWMAN S., DAS D., PAVLICK E.: What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations* (2019). 2

[VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer graphics forum 30*, 6 (2011), 1681–1707. 2

[VMV*21] VIG J., MADANI A., VARSHNEY L. R., XIONG C., RICHARD SOCHER, RAJANI N.: {BERT}ology meets biology: Interpreting attention in protein language models. In *International Conference on Learning Representations* (2021). 2

[VRM*17] VAROL G., ROMERO J., MARTIN X., MAHMOOD N., BLACK M. J., LAPTEV I., SCHMID C.: Learning from synthetic humans. In *CVPR* (2017). 4

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L., POLOSUKHIN I.: Attention is all you need. In *Proc. of NeurIPS* (2017). 2, 3, 8, 9

[VST21] VOITA E., SENNRICH R., TITOV I.: Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, Aug. 2021), Association for Computational Linguistics, pp. 1126–1140. 2

[VTM*19] VOITA E., TALBOT D., MOISEEV F., SENNRICH R., TITOV I.: Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 5797–5808. 2

[WML*20] WANG Y., MOHAMED A., LE D., LIU C., XIAO A., MAHADEOKAR J., HUANG H., TJANDRA A., ZHANG X., ZHANG F., FUEGEN C., ZWEIG G., SELTZER M. L.: Transformer-based acoustic modeling for hybrid speech recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), 6874–6878. 2

[YSI20] YOU W., SUN S., IYYER M.: Hard-coded Gaussian attention for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 7689–7700. 8, 10

[ZHLL20] ZHANG S., HUANG H., LIU J., LI H.: Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 882–890. 2

[ZKJB17a] ZUFFI S., KANAZAWA A., JACOBS D., BLACK M. J.: 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 5

[ZKJB17b] ZUFFI S., KANAZAWA A., JACOBS D. W., BLACK M. J.: 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 6365–6373. 5

[ZWC22] ZENG J., WANG D., CHEN P.: A survey on transformers for point cloud processing: An updated overview. *IEEE Access 10* (2022), 86510–86527. 2, 4