Kapszula alapú autoenkóder hálózatok rekonstrukciós hatékonysági vizsgálata pontfelhők esetén

Reconstruction efficiency analysis of capsule-based autoencoder networks for point clouds

Hollósi János[a] , Ballagi Áron[b] , Pozna Claudiu Radu[c]

[a]Széchenyi István Egyetem
hollosi.janos@sze.hu

[b]Széchenyi István Egyetem
ballagi@sze.hu

[b]Széchenyi István Egyetem
pozna@sze.hu

Absztrakt: A kapszulahálózatok elméletét és a kapszulák dinamikus útválasztási mechanizmusát Geoffrey Hinton és kutatócsoportja vezette be. Ebben az új megközelítésben a klasszikus konvolúciós neurális hálózatok tipikus problémáit próbálták megoldani. Például azt, hogy a neurális hálózatok hatékonysága romlik, ha a bemeneti képen geometriai transzformációt alkalmaznak, vagy ha az adatok messze vannak a képzési adathalmaztól. Már korán világossá vált, hogy a kapszula hálózatok a legkorszerűbb megoldások a vizuális adatok osztályozási feladataihoz. Viszont más feladatokra kevésbé elterjedt és sok esetben nehezen alkalmazható a használatuk. Ilyen például a képszegmentálás vagy a tárgyak felismerése és lokalizálása. A kapszulahálózatok elméletének hatékonysága a pontfelhő-feldolgozás területén is nyitott kérdés. Ebben a munkában a kapszulahálózatok pontfelhő-rekonstrukciós képességét vizsgáltuk. Ebben a megközelítésben három különböző komplexitású autoenkóder hálózatot választottunk ki. Létrehoztunk egy kapszula-elméleten alapuló dekódoló hálózatot, amelyet a meglévő enkóder hálózatokhoz illesztettünk. A hálózatok hatékonyságát négy különböző adathalmazon teszteltük. Munkánk eredményeként bemutatjuk a kapszulahálózatok hatékonyságát a pontfelhő-rekonstrukció területén a kiválasztott autoenkóder hálózatokkal összehasonlítva.

Kulcsszavak*:* kapszula hálózat, pontfelhő, autoenkóder, rekonstrukció

Abstract: The theory of capsule networks and the dynamic routing mechanism for capsules was introduced by Geoffrey Hinton and his research team. In this new approach, they tried to solve typical problems of classical convolutional neural networks. For example, that the

efficiency of neural networks degrades when a geometric transformation is applied on the input image, or when the data is far away from the training dataset. It became clear early on that capsule networks are state-of-the-art solutions for visual data classification tasks. For other tasks their use is less common and in many cases difficult to apply. For example image segmentation or object detection and localization. The efficiency of the capsule networks theory in the field of pointcloud processing is also an open question. In this work we investigated the pointcloud reconstruction capability of capsule networks. In this approach, three different complexity autoencoder networks was selected. We created a decoder network based on capsules theory, which was fitted to the existing autoencoder networks. The efficiency of the networks was tested using four different datasets. As a result of our work, we show the effectiveness of capsule networks in the field of pointcloud reconstruction compared with the selected autoencoder networks.

Keywords: capsule network, pointcloud, autoencoder, reconstruction

Bevezetés

A pontfelhőket egyre szélesebb körben használják a legkülönbözőbb területeken, például a számítógépes látás, a robotika és az autonóm járművek területén. Napjainkban egyre népszerűbb a pontfelhők neurális hálózatokkal történő feldolgozása. Például objektumosztályozás, objektumszegmentálás, objektumorientáció-érzékelés vagy objektumrekonstrukció. Ebben a munkában pontfelhők rekonstrukciójával foglalkozunk. A gyakorlatban használt pontfelhők meglehetősen nagyok, nehezen tárolhatók, továbbíthatók és feldolgozhatók. A kapszulahálózatok elmélete egy új terület a neurális hálózatokon belül, ahol a kapszulák felhasználhatósága a pontfelhő-rekonstrukció területén még felfedezetlen terület. Munkánkban egy kapszula alapú autoenkóder hálózat létrehozására teszünk kísérletet, amelynek rekonstrukciós képességét a klasszikus autoenkóder hálózatokkal hasonlítjuk össze.

Kapszula hálózatok elmélete

A kapszula hálózatok [1, 2] elméletének kidolgozása Geoffrey Hinton nevéhez köthető, aki sok más alapvető elmélet és algoritmus kidolgozója a mesterséges intelligencia területén belül. Az új elméletet a hagyományos neurális hálózatok nyomán alkották, ugyanakkor igyekeztek olyan új módszert létrehozni, amely a felvázolt problémák esetén robusztusabb, mint a neurális hálózatok. Amíg a neurális hálózatok építőeleme a neuron, addig a kapszula hálózatok

alapegysége az úgynevezett kapszula. A fő különbség a neuron és a kapszula között, hogy a neuron skalár értékekkel dolgozik, a kapszula pedig tetszőleges dimenziójú vektorokat használ. Úgy is tekinthetünk a kapszulára, mintha neuronok egy zárt, egybefüggő csoportját képeznék. A hálózatok esetén értelmezett különféle műveletek pedig ennek megfelelően módosulnak.

Legyen i egy kapszula és j egy i-nél magasabb szintű kapszula. Ekkor legyen a súlyozott bemenet

$$\hat{u}_{j|i} = \boldsymbol{W}_{ij} u_i \tag{1}$$

ahol $\boldsymbol{W}_{ij}$ a súlymátrix és u_i a bemeneti vektor az i-edik kapszula esetén.

Az úgynevezett összekapcsolási együttható megadása a softmax függvény segítségével történik a következők szerint

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{2}$$

ahol b_{ij} annak a valószínűsége, hogy az i-edik kapszula a j-edik kapszula irányába továbbítja a kimenetét. Az összegzett bemenet a j-edik kapszula számára az alábbiak szerint adható meg

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{3}$$

Kapszula hálózatok esetén az előrejelzés számszerűsítésére a kimeneti vektorok hosszát alkalmazzák. Ehhez az is szükséges, hogy a kapott vektorok hossza a $[0, 1]$ intervallumba kerüljön. Ehhez az elmélet kidolgozói bevezették az úgynevezett squash függvényt, amit a következő módon számíthatunk

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{4}$$

Az egymásra épülő kapszulák közötti információ áramlását a c súlyértékekkel lehet szabályozni. Ezeknek a számítása egy iteratív módon működő algoritmus segítségével történik. Ez az úgynevezett dinamikus útválasztás. Az útválasztási algoritmus célja, hogy a magasabb szintű és az alacsonyabb szintű kapszulák közötti információ közvetítés fő irányát meghatározza. Az útválasztó algoritmus működését az 1. ábra foglalja össze.

$$\forall i, j\ b_{ij} = 0$$
$$r\ iteráción\ keresztül:$$
$$\forall i\ c_i = \frac{e^{b_i}}{\sum_j e^{b_j}}$$
$$\forall j\ s_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i}$$
$$\forall j\ v_j = \frac{\|s\|^2}{1+\|s\|^2}\frac{s}{\|s\|}$$
$$\forall i, j\ b_{ij} = b_{ij} + \hat{\boldsymbol{u}}_{j|i}\boldsymbol{v}_j$$
$$ki: \boldsymbol{v}_j$$

1. ábra: Dinamikus útválasztás algoritmusa [1]

Felhasznált autoenkóder hálózatok

Munkánk során három különböző összetettségű és összetételű autóenkóder hálózatot vizsgáltunk. Az első Achlioptas és társai megoldása. [3] Ebben a megközelítésben minden ponthoz egy-egy konvolúciós réteg tartozik, ahol a kernel mérete 1. A konvolúciók után egy max pooling réteg elhelyezésével közös reprezentációt állítanak elő. Minden egyes konvolúciós réteget egy ReLU és egy Batch Normalization blokk követ. Mindezek után egy úgynevezett feature-wise maximum réteg egy k-dimenziós vektort állít elő, ahol $k = 512$ a mi implementációnkban. A dekódoló szakasz 3 teljesen összekapcsolt rétegen alapul, ahol az első kettő egy-egy ReLU aktiváló réteget is tartalmaz, amely egy 2048×3 méretű kimeneti vektort állít elő.

A következő megközelítésben Yang és társai [4] bemutatták a foldingnet architektúrát, ahol az enkóder egy gráf-alapú hálózat. Az enkóder blokk többrétegű perceptronokat (MLP) és gráfalapú max pooling rétegeket tartalmaz. Ez a megközelítés egy K-NNG gráfot használ a dekóder hálózatban. A dekódoló egy rögzített 2D rácsot használ, és megpróbálja ezt a rácsot egy kétlépcsős csomagolási mechanizmusban a bemeneti pontfelhő alakjára csomagolni. A rács mérete 45×45 pont, mivel ez a méret áll a legközelebb a bemeneti pontfelhő méretéhez, amely 2048 darab térbeli pontot tartalmaz.

Az utolsó felhasznált hálózat Pang és társai [5] munkája. Ebben a megközelítésben egy PointNet [6] alapú enkódert használtak. Ez egy rugalmas megoldás, mivel ez az enkóder topológiabarát reprezentációkat tud generálni. A dekóder 3 szakaszból épül fel: az első egy FoldingNet, a következő egy TearingNet, az utolsó pedig ismét egy FoldingNet. Ez a

megközelítés egy primitív gráf éleit megtörve, a gráfot foltokra vagy lyukakra tépi, hogy a célpontfelhő topológiáját utánozza.

Megvalósított kapszula dekóder

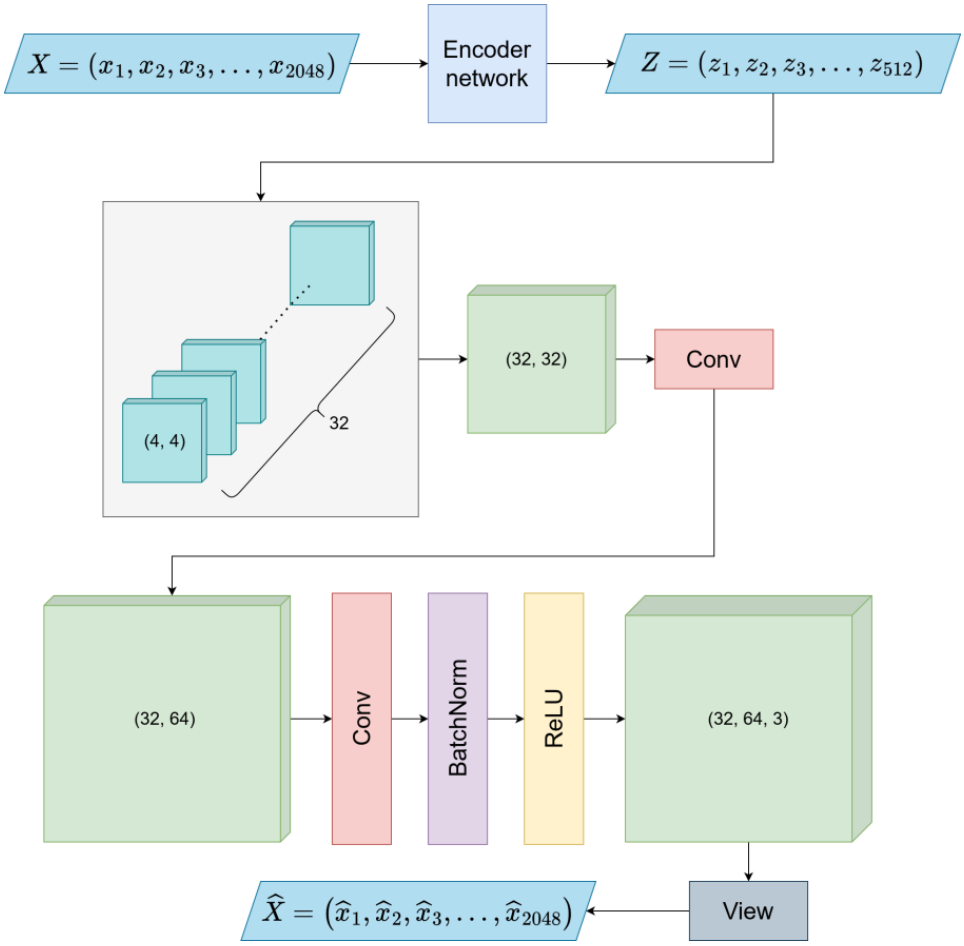
A kapszulahálózatok elméletén alapuló dekódoló modellt hoztunk létre. Az általunk létrehozott dekódoló modellt a 3. fejezetben bemutatott három autoenkóder hálózat enkóder szakaszával egyesítettük, tanítottuk és teszteltük. A modellünk bemenete legyen a következő:

$$Z = (z_1, z_2, \dots, z_{512}) \tag{5}$$

ahol Z a hálózat enkóder részének kimeneti vektora, 512 darab jellemzőponttal. Modellünk első része egy kapszulablokk dinamikus útválasztó algoritmussal. Ez a blokk 32 kapszulát tartalmaz, ahol minden kapszula alakja $(4, 4)$. A dinamikus útválasztási algoritmusban fix számú útvonalat használunk. Sabour és társai [1] alapján 3 iteráción keresztül végezzük a pontosítást. Ennek a kapszulablokknak a kimeneti tenzor alakja $(32, 32)$. Ahol egy egyszerű konvolúciós blokkot alkalmazunk, hogy a tenzor méretét $(32, 64)$-re növeljük. Az utolsó lépések a konvolúció, a normalizálás, majd egy lineáris aktivációs függvény. A kimeneti tenzor alakja $(32, 64, 3)$. Majd ezt transzformáljuk át $(2048, 3)$ alakúra:

$$\hat{X} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{2048}) \tag{6}$$

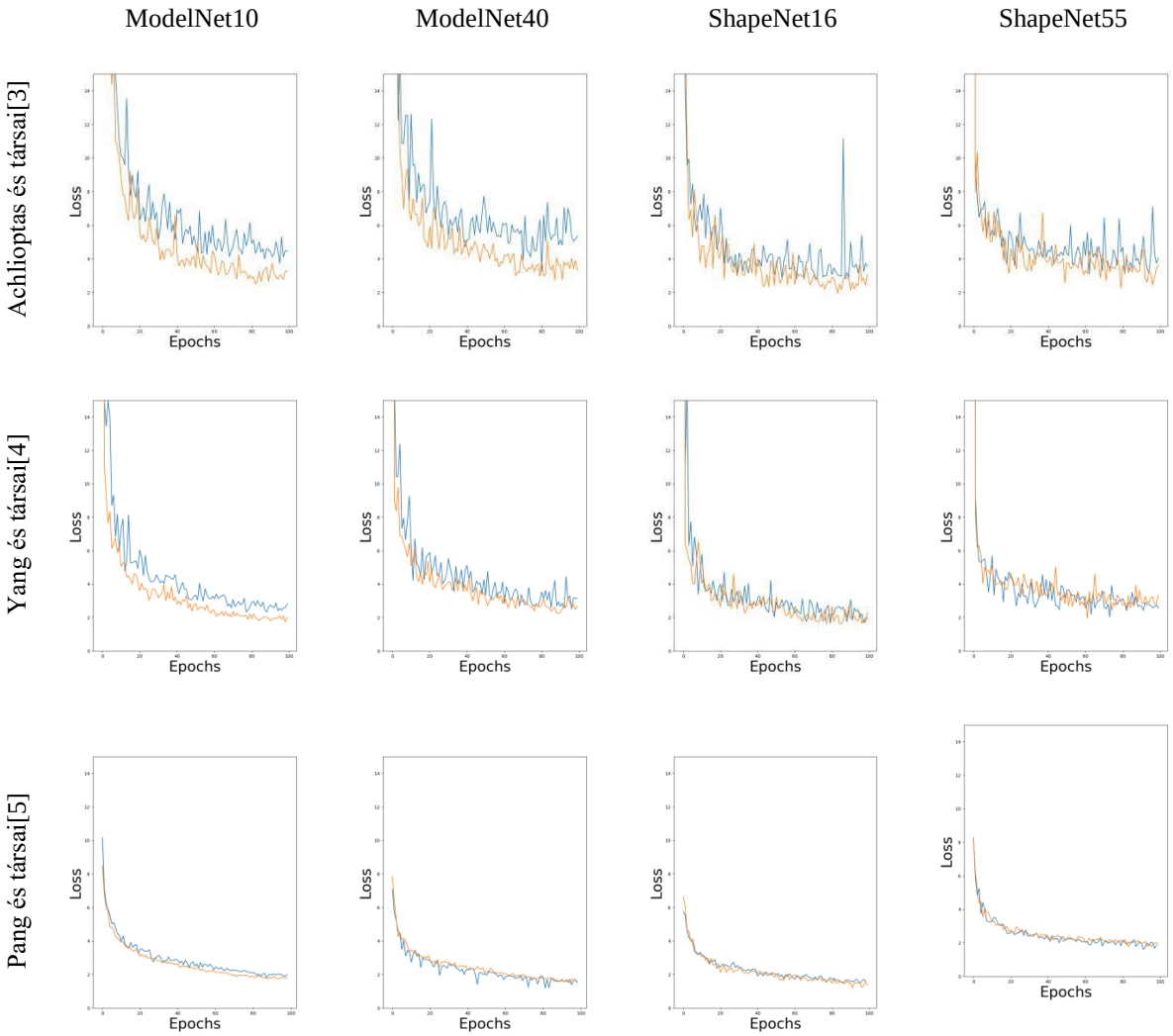
ahol $\hat{X}$ vektor a hálózatunk kimenete, amely a pontfelhő rekonstruált pontjainak halmazát tartalmazza. Az 2. ábra mutatja az ebben a fejezetben bemutatott kapszula alapú dekóder modell felépítését.



2. ábra: Kapszula alapú dekóder hálózat blokkvázlata

Adatkészletek

Ebben a munkában két népszerű adatkészletet használtunk: ModelNet [7] és ShapeNet [8]. Ezek az adatkészletek számos kategóriába tartozó különböző objektumok CAD-modelljeit tartalmazzák. Mindkét adatkészlethez egy egyszerűsített adathalmazt is generáltunk. Így összesen négy adathalmazt használtunk: ModelNet10, ModelNet40, ShapeNet16, ShapeNet55. A ModelNet10 adatkészletet a ModelNetből generálták. Ez 4899 alakzatot tartalmaz 10 tárgykategóriából. A tárgykategóriák a következők: fürdőkád, ágy, szék, íróasztal, komód, monitor, éjjeliszekrény, kanapé, asztal és WC. A ModelNet40 a teljes ModelNet-adatkészlet 40 tárgykategóriával és 12311 alakzattal. A ShapeNet16 adathalmaz 16881 objektumot tartalmaz 16 kategóriában: repülőgép, táska, sapka, autó, szék, fülhallgató, gitár, kés, lámpa, laptop, motorkerékpár, bögre, pisztoly, rakéta, gördeszka és asztal. A legösszetettebb



3. ábra: Tanítás eredménye (kékkel a szerzők eredeti dekódere, sárgával a saját kapszula dekóderünk esetén)

adatkészlet a ShapeNet55, amely 51127 darab objektumot és a következő objektumkategóriákat tartalmazza: repülőgép, táska, kosár, fürdőkád, ágy, pad, madárház, könyvespolc, üveg, tál, busz, szekrény, fényképezőgép, konzervdoboz, sapka, autó, mobiltelefon, szék, óra, mosogatógép, fülhallgató, csap, fájl, gitár, sisak, korsó, billentyűzet, kés, lámpa, laptop, postaláda, mikrofon, mikrohullámú sütő, monitor, motorkerékpár, bögre, zongora, párna, pisztoly, cserép, nyomtató, távirányító, puska, rakéta, gördeszka, kanapé, hangszóró, kályha, asztal, telefon, konzervdoboz, torony, vonat, edény és mosógép.

Eredmények

A 3. fejezetben leírt 3 hálózatot használtuk ebben a tanulmányban. Minden autoenkóder hálózathoz létrehoztunk egy kapszula-alapú dekóder hálózat megoldást, ahol mindhárom hálózat esetén annak eredeti dekóderét a mi kapszula-dekóder megközelítésünkkel helyettesítettük. Munkánkban az eredeti hálózatokat hasonlítottuk össze a kapszula alapú megoldásokkal. Mind a 6 hálózatot az 5. fejezetben bemutatott adatkészletekkel képeztük ki. Minden hálózatot 100 iteráción keresztül treníroztunk, ahol az Adam [9] optimalizáló algoritmus került felhasználásra. Ebben a megközelítésben a tanulási ráta értéke 5×10^{-4}, 12-es kötegméret mellett. A Chamfer Distance (CD) [10] metrikát használtuk a bemeneti pontfelhő (X) és a rekonstruált pontfelhő ($\hat{X}$) közötti különbség kiszámítására a következők szerint:

$$CD(X, \hat{X}) = \sum_{x \in X} \min_{\hat{x} \in \hat{X}} \|x - \hat{x}\|_2^2 + \sum_{\hat{x} \in \hat{X}} \min_{x \in X} \|x - \hat{x}\|_2^2 \quad (7)$$

A 3. ábra a bemutatott konfiguráció szerinti tanítás eredményét szemlélteti minden hálózati architektúra esetében. Az ábrán a tanulás folyamatát láthatjuk, ahol veszteségfüggvényként (Loss) a Chamfer Distance metrika lett szemléltetve. Az I. táblázat a Chamfer Distance szerinti legjobb hatékonyságokat összegzi és hasonlítja össze. A Chamfer Distance metrika értéke akkor 0, ha a bemeneti és a kimeneti adat teljes egészében megegyezik. Tehát az érték minél inkább közelíti a 0-t, annál jobb eredményt mutat a hálózat.

Az eredmények azt mutatják, hogy a legtöbb esetben a kapszula alapú dekódoló megoldásunk jobban teljesített, mint a megfigyelt neurális hálózat alapú megközelítések. A kapszulaalapú megoldás 12 esetből 10 esetben jobb eredményt adtak, mint a hagyományos neurális hálózat alapú megvalósítások.

Megfigyelhető, hogy a legegyszerűbb hálózat esetében (Achlioptas és társai [3]) a kapszula alapú megoldás mindig jobban teljesített. A Yang és társai [4] megoldása már komplexebb, de a kapszula alapú megoldásunk még itt is mindig jobb kimenettel szolgált. A legösszetettebb megoldás a Pang és társai [5] módszere, ahol a kapszulás dekódoló már csak a két egyszerűbb adatkészlet esetén tudott hatékonyabb megoldást adni.

Meg kell jegyezni, hogy az általunk bemutatott kapszula alapú dekódoló hálózat komplexitása alacsony. Ezért nem biztos, hogy mindig jobb megoldást nyújt a bonyolultabb neurális hálózat

1. táblázat: Hálózatok hatékonysága a különféle adatkészleteken

	ModelNet10	ModelNet40	ShapeNet16	ShapeNet55
Achlioptas és tsai[3]	3,7346	3,2378	2,7896	3,0954
Saját megoldásunk	**2,4936**	**2,7500**	**1,9490**	**2,2465**
Yang és tsai[4]	2,3230	2,5099	1,6566	2,0187
Saját megoldásunk	**1,7244**	**2,2486**	**1,6014**	**1,9531**
Pang és tsai[5]	1,8104	**1,2012**	1,4819	**1,6082**
Saját megoldásunk	**1,7625**	1,4882	**1,2009**	1,7708

alapú megoldásokhoz képest. Az azonban látható, hogy egy egyszerű kapszula alapú megoldás is felveheti a versenyt a komplex neurális hálózat alapú megoldásokkal a pontfelhő-rekonstrukció területén. A jövőben szeretnénk komplexebb kapszula-alapú megoldásokat vizsgálni, hogy kiaknázzuk a kapszulahálózatokban rejlő lehetőségeket.

Köszönetnyilvánítás

A TKP2021-NKTA-48 számú projekt az Innovációs és Technológiai Minisztérium Nemzeti Kutatási, Fejlesztési és Innovációs Alapból nyújtott támogatásával, a TKP2021-NKTA pályázati program finanszírozásában valósult meg.

Irodalomjegyzék

[1] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic routing between capsules,” Advances in Neural Information Processing Systems, 2017.

[2] G. E. Hinton, S. Sabour and N. Frosst, “Matrix capsules with em routing,” Sixth International Conference on Learning Representations, 2018.

[3] P. Achlioptas, O. Diamanti, I. Mitliagkas and L. Guibas, “Learning Representations and Generative Models for 3D Point Clouds,” arXiv: 10.48550/ARXIV.1707.02392

[4] Y. Yang, C. Feng, Y. Shen and D. Tian, “FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 206-215, doi: 10.1109/CVPR.2018.00029.

[5] J. Pang, D. Li and D. Tian, “TearingNet: Point Cloud Autoencoder to Learn Topology-Friendly Representations,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7449-7458, doi: 10.1109/CVPR46437.2021.00737.

[6] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[7] Zhirong Wu et al., “3D ShapeNets: A deep representation for volumetric shapes, ” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912-1920, doi: 10.1109/CVPR.2015.7298801.

[8] M. Savva, A. X. Chang and P. Hanrahan, "Semantically-enriched 3D models for common-sense knowledge," 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 24-31, doi: 10.1109/CVPRW.2015.7301289.

[9] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” In Int. Conf. Learn. Represent., 2015.

[10] H. Fan, H. Su and L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2463-2471, doi: 10.1109/CVPR.2017.264.