

# Projekt naukowo-wdrożeniowy

## GenWro3D

Bilski Kajetan, Chodukiewicz Maciej, Marciniak Michał, Woźniak Stanisław  
Opiekun: dr hab. inż. Maciej Zięba

26 czerwiec 2022r.

## 1 Wstęp oraz cel projektu

Drzewo *CSG* (ang. *Constructive solid geometry*) jest sposobem reprezentowania skomplikowanych obiektów jako zbioru prymitywnych kształtów oraz operacji między nimi. Metoda ta ma zastosowanie przykładowo w branży *Game Development*, gdzie służy do tworzenia tzw. *colliderów*, czyli uproszczonych modeli obiektów, które pozwalają na łatwiejsze, a co za tym idzie szybsze obliczanie fizyki obiektów.

Celem naszego projektu było udoskonalenie istniejących metod do tworzenia drzew *CSG*. Udało się to osiągnąć poprzez zmniejszenie wielkości drzewa poprzez zmniejszenie liczby prymitywów niezbędnych do wykonania rekonstrukcji, bez znaczącej straty jakości rekonstrukcji oraz stworzenia modelu generatywnego, który umożliwia próbkowanie nowych obiektów wraz z ich drzewami *CSG* z wyuczonej przestrzeni ukrytej.

## 2 Udoskonalenie UCSG-Net

Jako bazowy model do generowania obiektów i drzew użyliśmy model UCSG-Net [2]. Model ten rekonstruował obiekt wejściowy predykując jednocześnie drzewo *CSG* tego obiektu. Jednakże UCSG-Net posiadał pewne problemy. Jednym z takich problemów było predykowanie drzew, które nie były optymalne. Oznacza to, że prymitywne kształty, które zostały wypredykowane przez model oraz z nich zostało zbudowane drzewo *CSG*, były w wielu przypadkach nieużyteczne i nic nie wносиły do końcowej rekonstrukcji. Skutkiem tego były duże drzewa.

Na tym etapie projektu, naszym celem było zmodyfikowanie i udoskonalenie sieci UCSG-Net, aby wyjściowe drzewa *CSG* posiadały mniejszą liczbę bezużytecznych kształtów. Zastosowaliśmy dwie metody modyfikowania modelu.

Oryginalnie UCSG-Net predykował jako prymitywy jedynie prostokąty oraz koła. W celu zmniejszenia liczby potrzebnych kształtów dodaliśmy kolejny rodzaj prymitywów: trójkąty. W tym celu dodaliśmy do architektury moduł, który był odpowiedzialny za tworzenie trójkątów.

Inną modyfikacją było użycie uczenia nadzorowanego na prymitywy. Trzeba tu podkreślić, że etykietami danych był jedynie zbiór prymitywów, z których te obiekty się składały a nie całe, wyjściowe drzewo. Dlatego model łączył te kształty wciąż w sposób nienadzorowany.

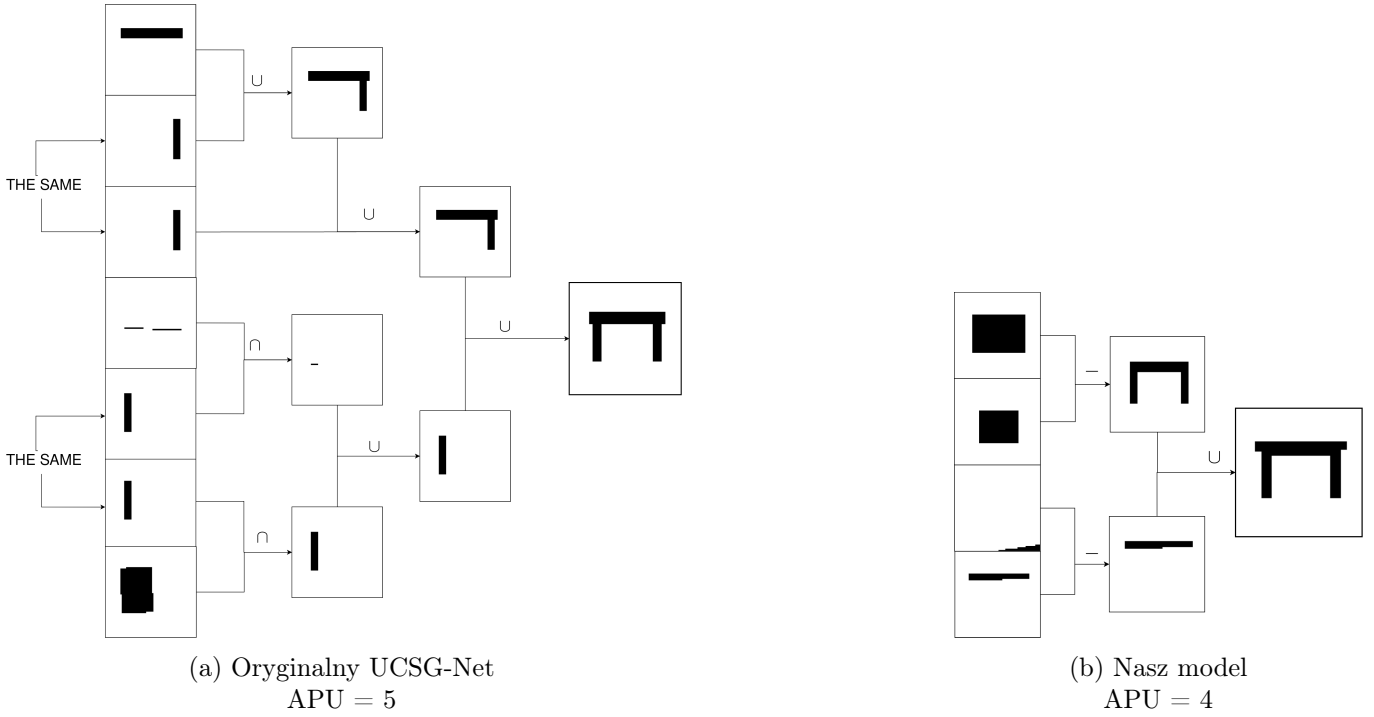
W tabeli 1 zostały pokazane wyniki naszych modeli w porównaniu do oryginalnego UCSG-Net. Model *Unsup 2-6* jest modelem w którym jedynie zostały dodane trójkąty. Natomiast modele *Reg 2-4* oraz *Reg 3-6* są to modele z trójkątami oraz uczeniem nadzorowanym na prymitywach, ale z różnymi hiperparametrami. W naszych badaniach i eksperymentach zostały zbadanych również wiele innych, jednakże te dawały najlepsze wyniki.

Wyniki zostały porównane za pomocą 3 metryk: Intersection over Union (IoU), którą chcemy maksymalizować, Chamfer Distances (CD), którą chcemy minimalizować, oraz Average Primitives Use (APU), którą chcemy minimalizować. Ta ostatnia metryka jest metryką do mierzenia wielkości drzewa. Obecnie w dziedzinie, nie istnieje metryka do mierzenia jakości drzew *CSG*, dlatego sami zaprezentowaliśmy metrykę APU, która określa liczbę różnorodnych prymitywów w drzewie. Innym słowem jest to liczba unikalnych liście w drzewie *CSG*. Natomiast metryki IoU oraz CD są metrykami jakości rekonstrukcji.

Nazwa	Zbiór treningowy			Zbiór walidacyjny			Zbiór testowy		
	IoU	CD	APU	IoU	CD	APU	IoU	CD	APU
Oryginalny UCSG-Net	<b>0.91</b>	<b>0.29</b>	5.69	<b>0.9</b>	<b>0.33</b>	5.69	<b>0.91</b>	<b>0.3</b>	5.69
Unsup 2-6 (nasz)	0.85	0.59	5.17	0.83	0.62	5.16	0.84	0.6	5.18
Reg 2-4 (nasz)	0.87	0.45	<b>4.53</b>	0.86	0.47	<b>4.54</b>	0.87	0.45	<b>4.52</b>
Reg 3-6 (nasz)	0.89	0.39	10.82	0.88	0.42	10.79	0.88	0.39	10.82

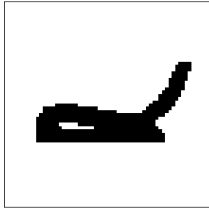
Tablica 1: Wyniki naszych modeli w porównaniu do oryginalnego UCSG-Net

Nasze modele w przypadku jakości rekonstrukcji nie poprawiły wyników w porównaniu do bazowego ucSG-Net, jednakże *Reg 2-4* z bardzo podobną jakością potrafi zrekonstruować obiekty za pomocą mniejszego drzewa. Przykładowe porównanie na Fig. 1.

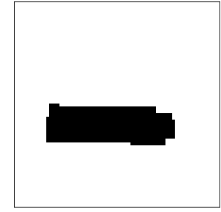


Rysunek 1: Porównanie drzew CSG

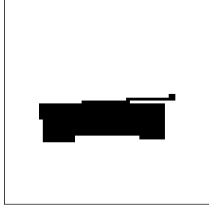
Natomiast modele *Unsup 2-6* oraz *Reg 3-6* potrafią lepiej zrekonstruować obiekty, których nie umie poprawnie zrekonstruować ani oryginalny UCSG-Net ani *Reg 2-4*. Na Fig. 2 przykładowe porównanie.



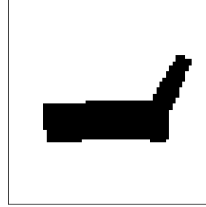
(a) Oryginalny obiekt



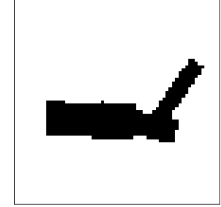
(b) UCSG-Net



(c) Reg 2-4



(d) Reg 3-6



(e) Unsup 2-6

Rysunek 2: Przykład porównania rekonstrukcji

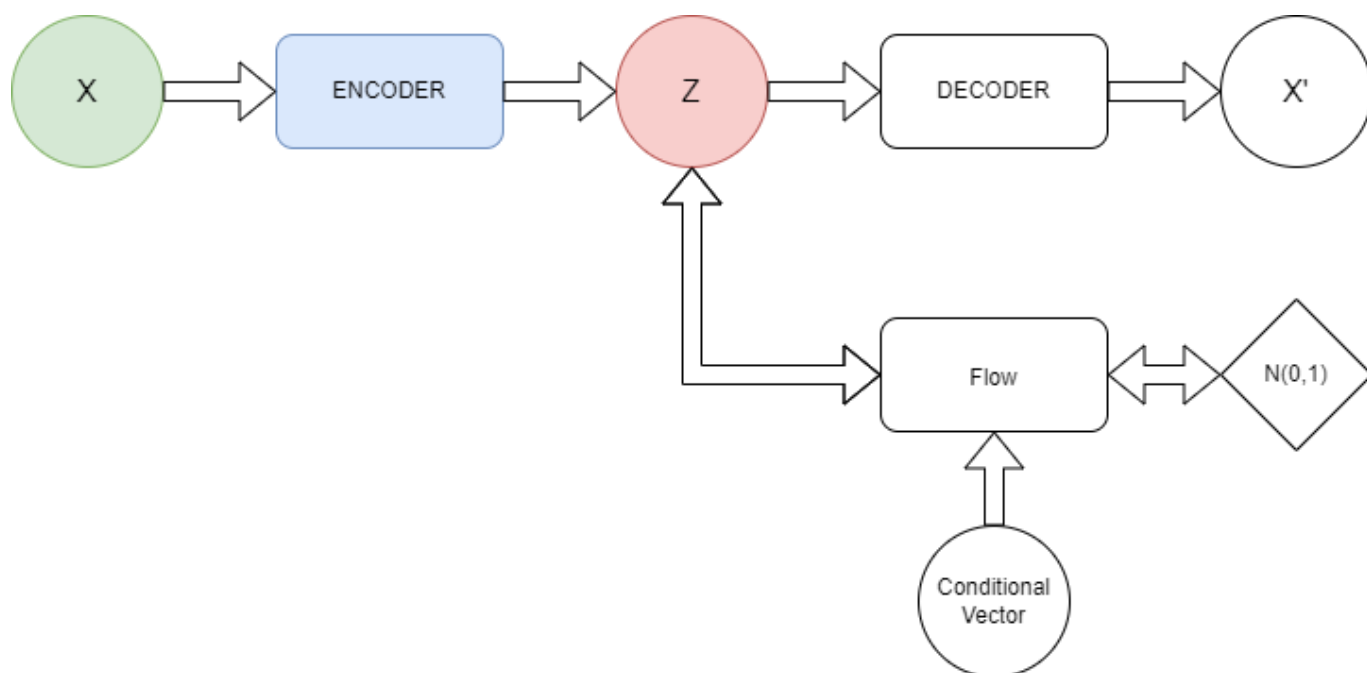
### 3 Model generatywny

Dotychczasowe podejście opierało się na znajdowaniu drzewa CSG dla danego obiektu. Nie znaleźliśmy podejścia, które umożliwiałoby generowanie takich drzew, bez konieczności zadawania obiektu docelowego. W związku z tym opracowaliśmy metodę umożliwiającą rozszerzenie bazowego modelu o zdolności generatywne. W tym celu zastosowaliśmy przepływy normalizacyjne jako "wtyczkę" podpinaną pod przestrzeń ukrytą utworzoną przez enkoder. W celu wytrenowania takiej wtyczki należy przygotować dane wejściowe. W tym celu wykorzystano zamrożony enkoder, który zakodował próbki z zbioru danych do postaci wektorów osadzeń. Proces ten został zaprezentowany schematycznie na rys. 3. Sam trening odbywa się poprzez optymalizację metodą największej wiarygodności dla znanego rozkładu, np. Gaussowskiego. Trening został schematycznie zaprezentowany na rys. 4. Mając wytrenowaną wtyczkę możemy próbkować z rozkładu bazowego, a następnie transformować takie próbki do postaci wektorów osadzeń i przepuszczać je przez dekodery, generując nowe obiekty - zaprezentowane na rys. 5.

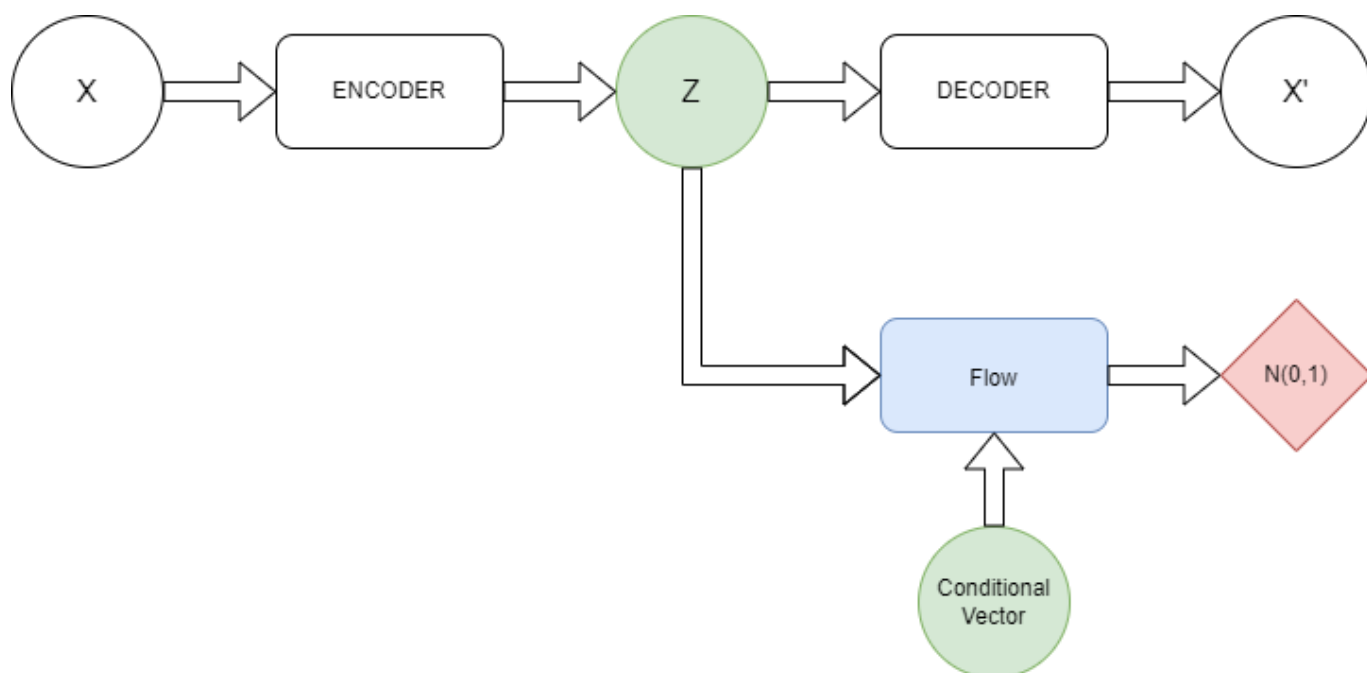
Jako model bazowy dla naszej wtyczki wykorzystaliśmy model RealNVP [1]. Zostały wytrenowane wtyczki dla 3 wariantów modeli z poprzedniego rozdziału:  $Reg_{2-4}$ ,  $Reg_{3-6}$  oraz  $Unsup_{2-6}$ . Wyniki zostały zaprezentowane w Tabeli 2. Jak można zauważyć, wszystkie modele wypadły podobnie pod kątem metryki minimum-matching-distance. Mówi nam to o tym, że generowały podobne pod względem jakości obiekty. Natomiast w przypadku metryki Coverage, można zauważyć lekką przewagę wtyczki wytrenowanej dla modelu bazowego trenowanego w sposób nienadzorowany. Świadczy to o większej różnorodności generowanych próbek. Na rys. 6 zaprezentowano przykładową przestrzeń ukrytą zamodelowaną przez enkoder oraz model generatywny. Jak widać model generatywny w postaci wtyczki opartej o przepływy normalizacyjne jest w stanie nauczyć się modelować przestrzeń odpowiadającą przestrzeni ukrytej generowanej przez enkoder. Świadczy to o zdolności do generowania nowych próbek zbioru danych za pomocą próbkowania z znanego rozkładu. Do redukcji wymiarowości została wykorzystana technika Umap [3]. Na rys. 7 zostały zaprezentowane przykładowe obiekty, których proces konstrukcji opisany został przez wygenerowane drzewa CSG.

Model	CD-MMD ( $\downarrow$ )	CD-COV ( $\%, \uparrow$ )
Reg 2-4	0.953	38.00
Reg 3-6	<b>0.893</b>	38.43
Unsup 2-6	0.899	<b>40.10</b>

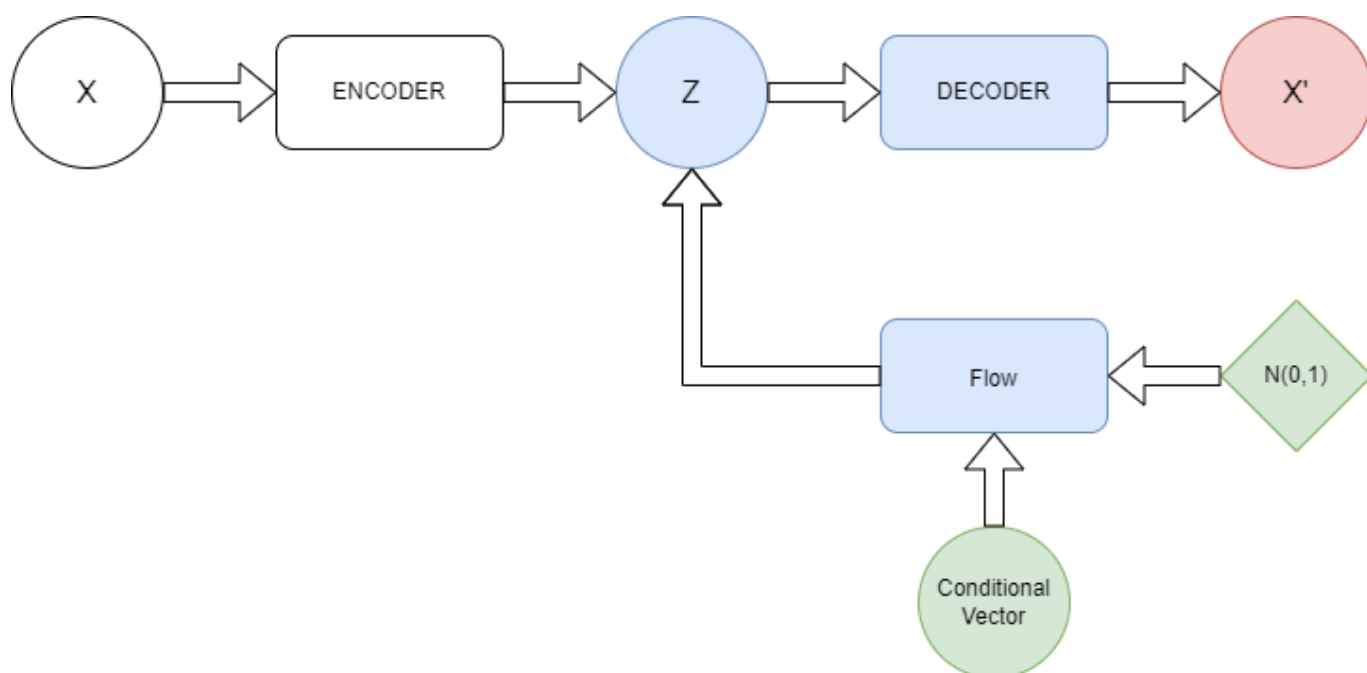
Tablica 2: Porównanie modeli generatywnych.  $\uparrow$ : Im wyżej tym lepiej.  $\downarrow$ : Im niżej tym lepiej.



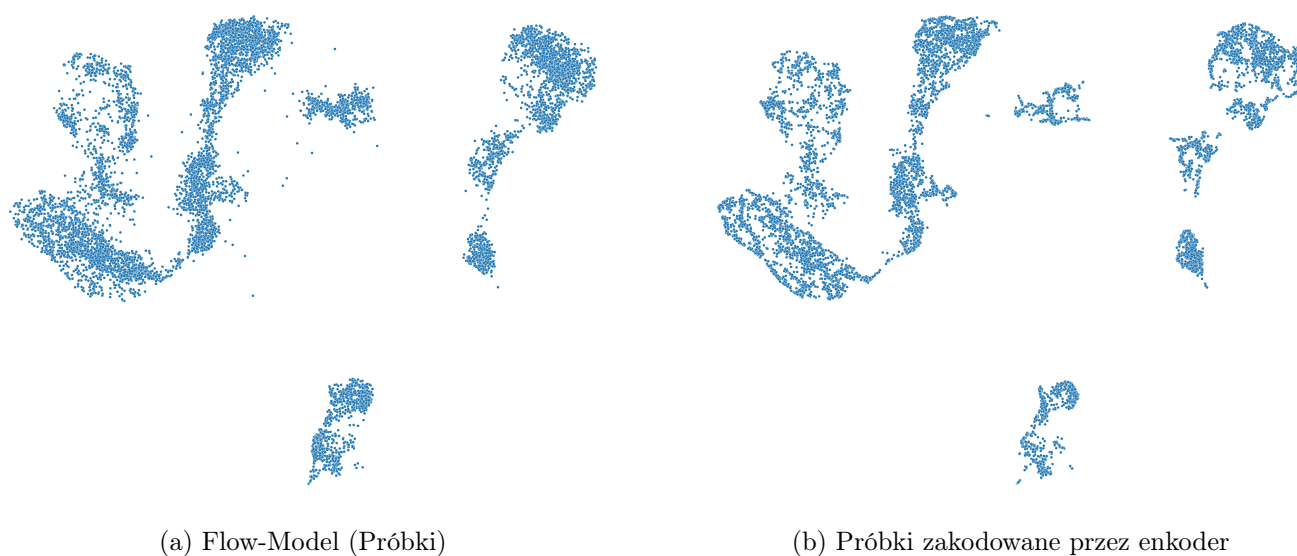
Rysunek 3: Schemat enkodowania danych do treningu flowa. Legenda: zielony - dane wejściowe, niebieski - kroki pośrednie, czerwony - dane wyjściowe.



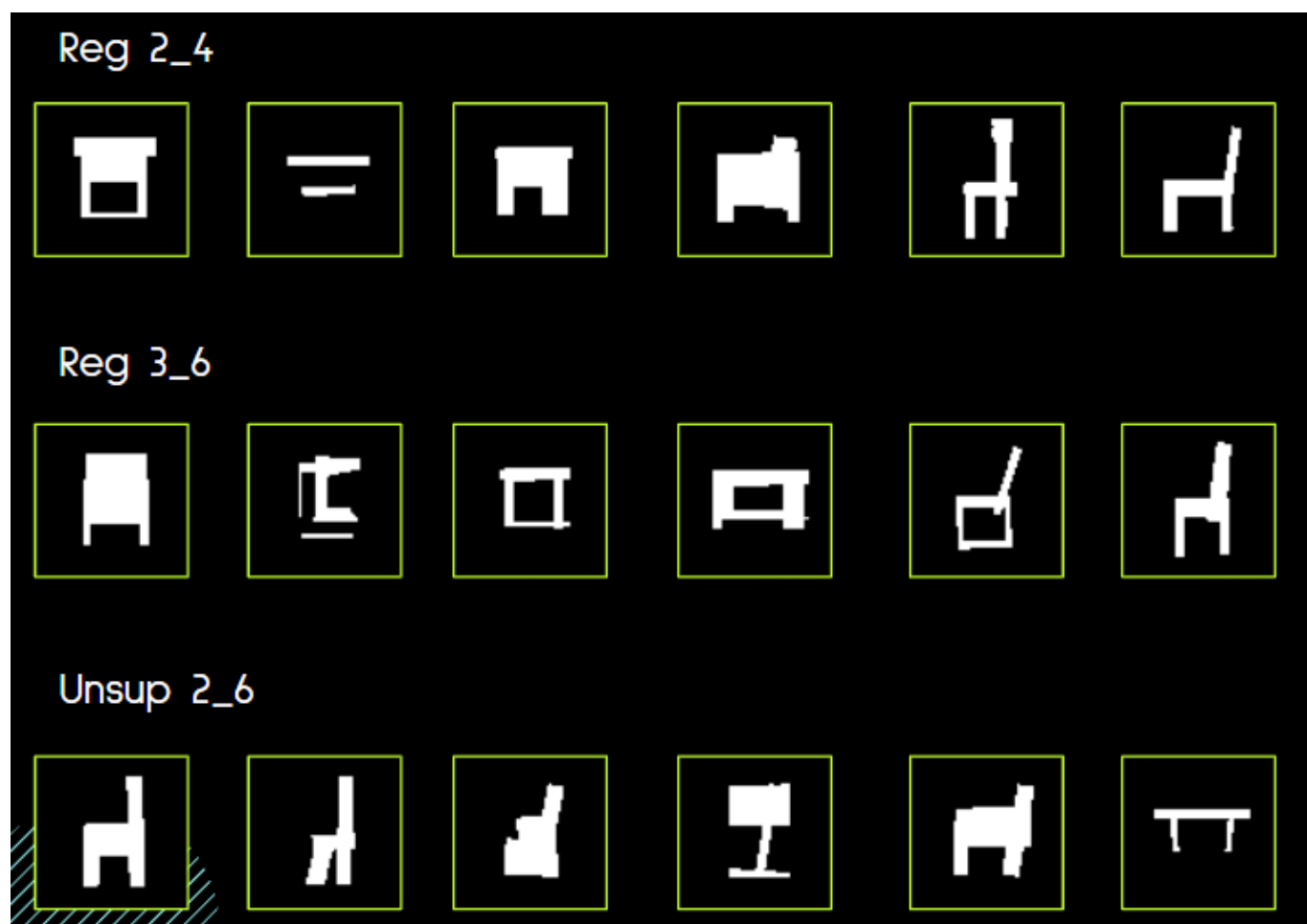
Rysunek 4: Schemat treningu wtyczki. Legenda: zielony - dane wejściowe, niebieski - kroki pośrednie, czerwony - dane wyjściowe.



Rysunek 5: Schemat generowania nowych próbek przy wykorzystaniu wytrenowanej wtyczki. Legenda: zielony - dane wejściowe, niebieski - kroki pośrednie, czerwony - dane wyjściowe.



Rysunek 6: Wizualizacja przestrzeni ukrytej utworzona za pomocą techniki UMAP [3].



Rysunek 7: Przykłady wyrenderowanych obiektów opisanych przez wygenerowane drzewa CSG.

## Literatura

- [1] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016.
- [2] K. Kania, M. Zieba, and T. Kajdanowicz. Ucs-g-net-unsupervised discovering of constructive solid geometry tree. *Advances in Neural Information Processing Systems*, 33:8776–8786, 2020.
- [3] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.