



神经搜索在 3D 场景中的研究与应用

王峰

Jina AI

About Me



2021-now, Senior AI Engineer, *Jina AI*
2020-21, Senior Researcher, *Huya AI*
2018-19, Senior Researcher, *Tencent AI*
2011-18, Ph. D., *Hong Kong Baptist University*

王峰, 开源神经搜索框架 [Jina](#) 的核心贡献者, 专注机器学习与深度学习算法在 NLP, 多模态表征学习和信息检索领域的落地与应用。



About Jina AI

- 👤 Founded in 2020/2
- 👥 53 members
- 🌐 4 offices
- 💰 Raised \$38M+
- 🏆 Top-tier AI company



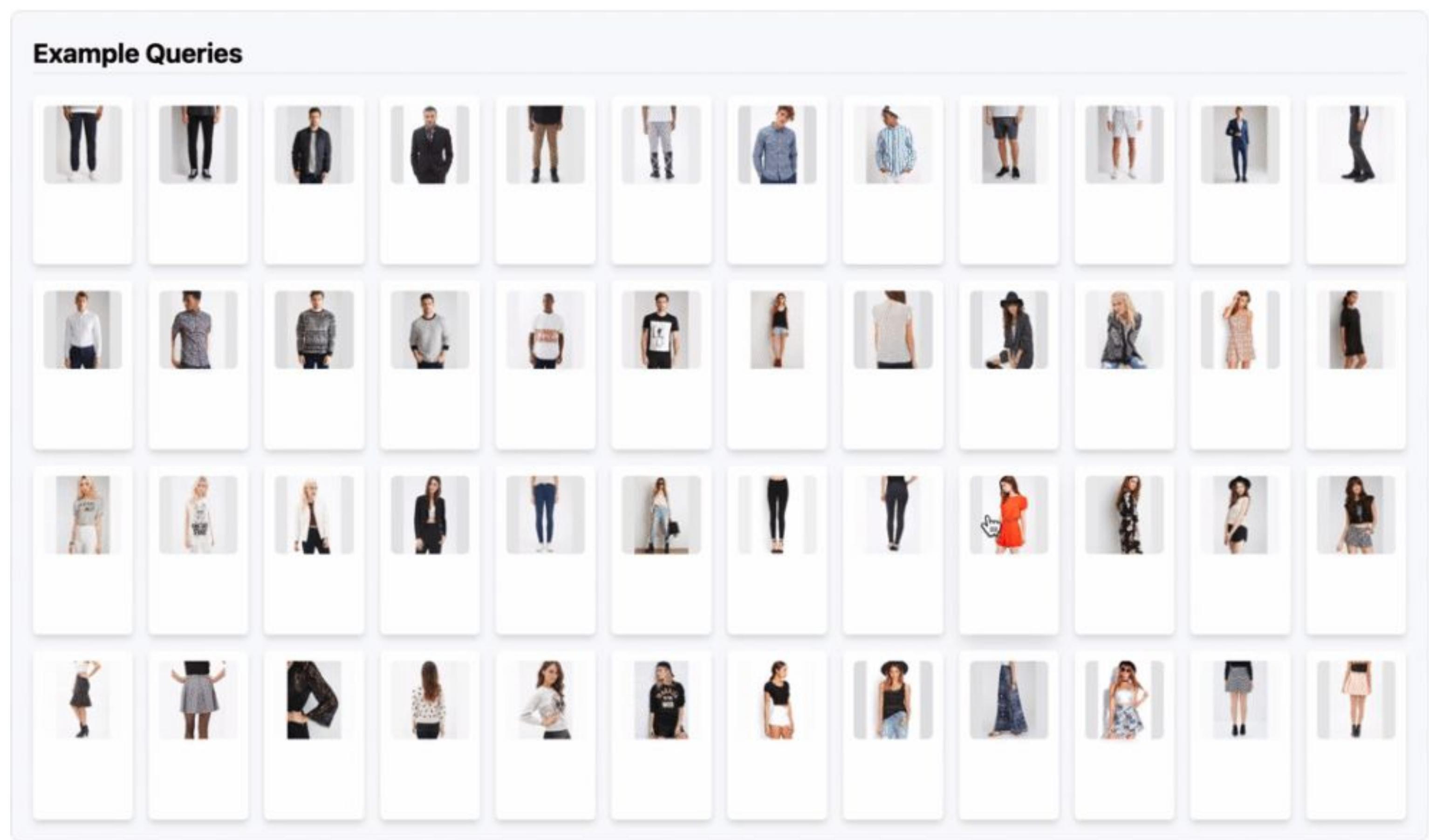
What is Neural Search?

01

What is Neural Search?

Find similar fashion items

Given a fashion item, find top-k most visually similar fashion items in the stock.



一起点亮
GitHub
欢迎大家 Star
★

What is Neural Search?

Question-answer chatbot

Answer arbitrary questions in a closed-domain.

astAPI

FastAPI framework, high performance, easy to learn, fast to code, ready for production

 pypi package v0.78.0  python 3.6 | 3.7 | 3.8 | 3.9 | 3.10

jolo/fastapi

FastAPI (Fast API) is a modern, fast (high performance), web framework for building APIs with Python 3.6+

FastAPI is built on top of Starlette (high performance), and it integrates with NodeJS and Go (thanks to Starlette and Pydantic).

Table of contents

- Sponsors
- Opinions
- Typer, the FastAPI
- Requirements
- Installation
- Example
- Create it
- Run it
- Check it
- Interactive API
- Alternative API
- Example upgrade
- Interactive API
- Alternative API
- Recap
- Performance
- Optional Dependencies
- License



FastAPI Bot

FastAPI framework, high performance, easy to learn, fast to code, ready for production

You can ask questions about FastAPI.

Try:

How do you deploy FastAPI?

What are type hints?

What is OpenAPI?

How do you deploy FastAPI?

When deploying FastAPI applications a common approach is to **build a Linux container image**.

[See context](#)

Type your question here...



一起点亮

GitHub

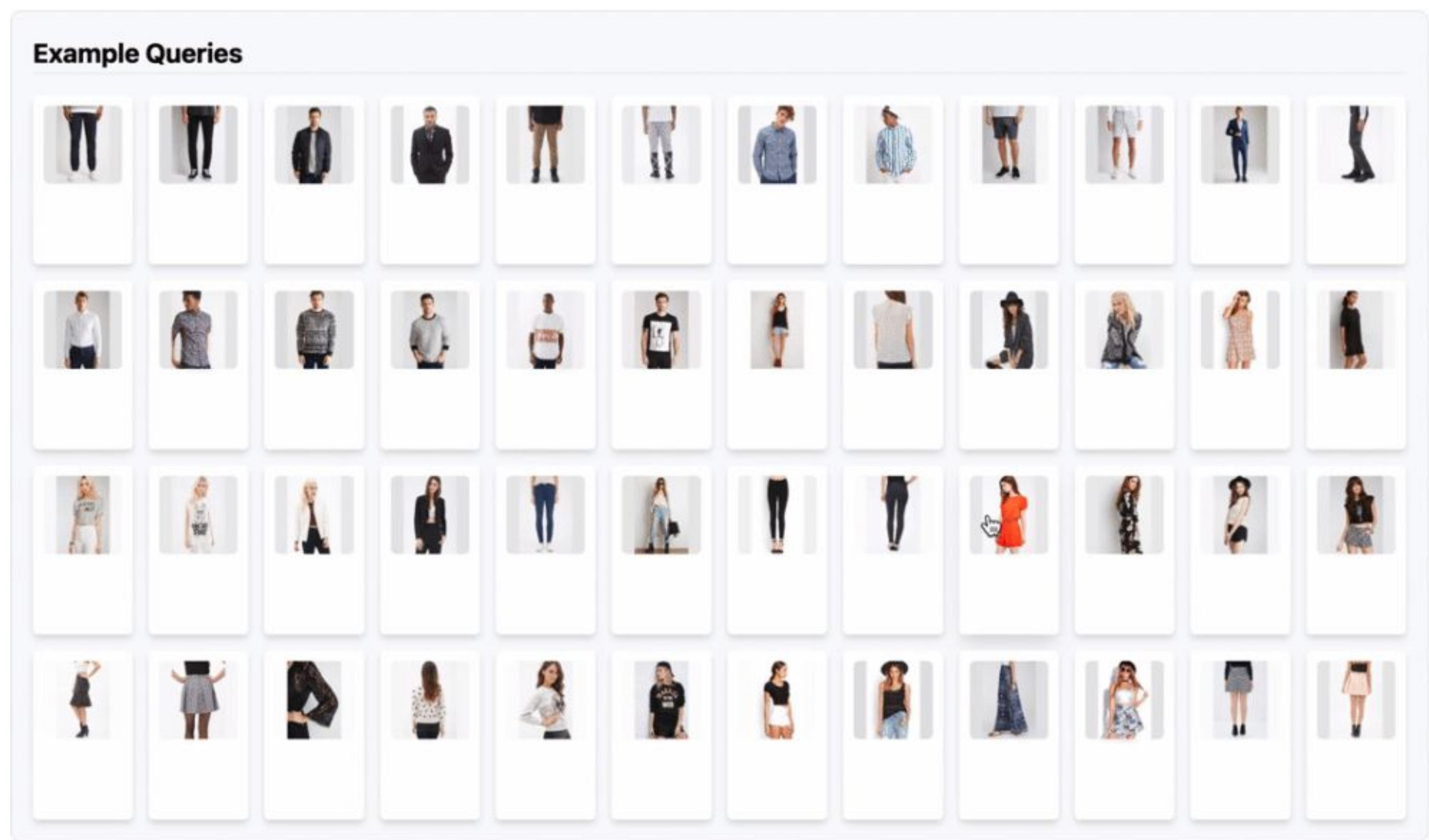
欢迎大家 Star



What is Neural Search?

Find similar fashion items

Given a fashion item, find top-k most visually similar fashion items in the stock.

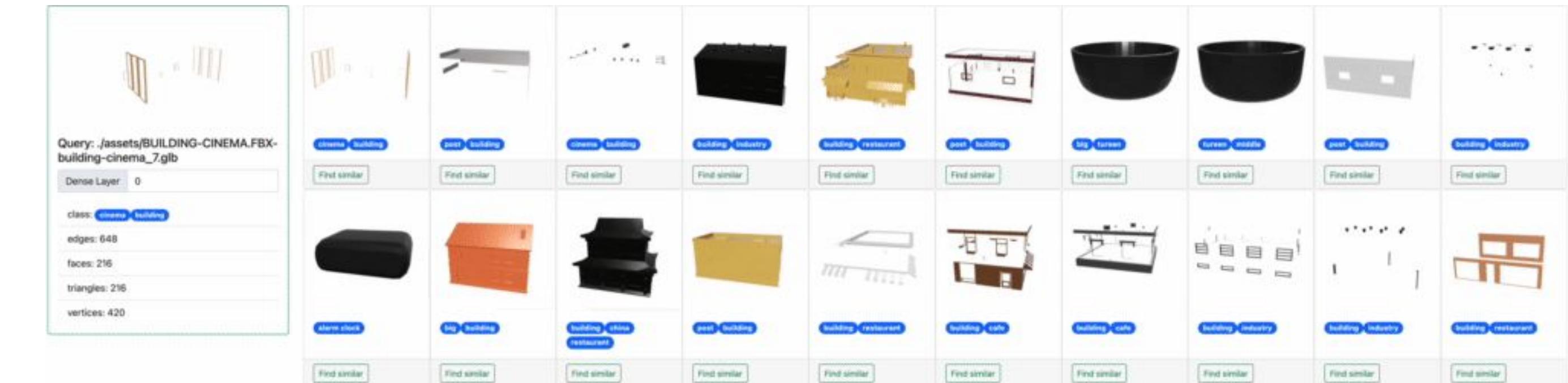
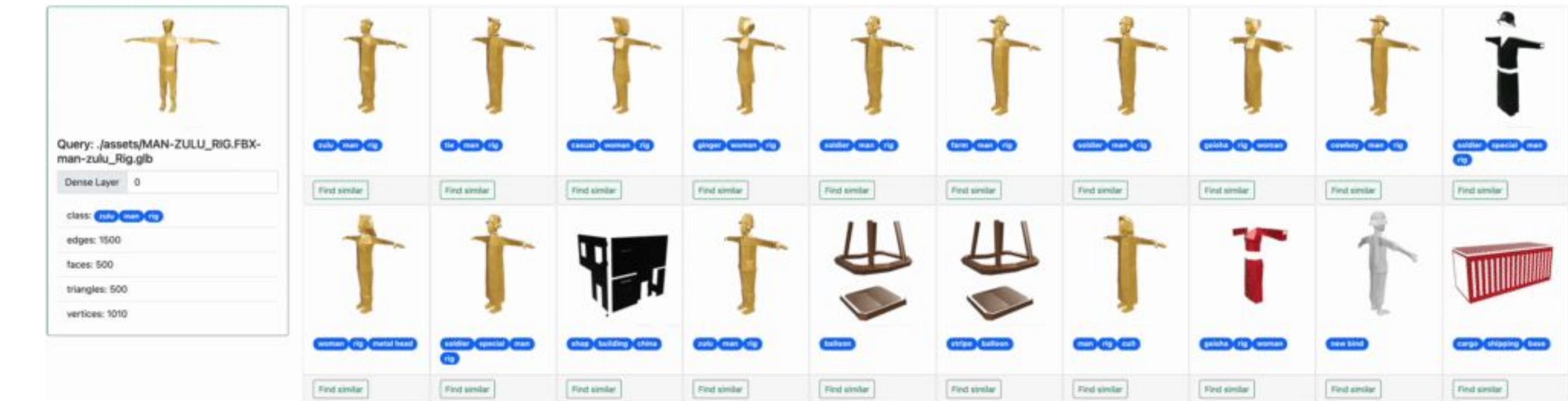


一起点亮
GitHub
欢迎大家 Star
★

What is Neural Search?

3D mesh look-like

Enable “look-like” feature in game development



Fundamental

Keyword Search vs Neural Search

Keyword Search Example

Query: What is Jina?

Result: Jina is a cloud-native neural search framework to build SOTA and scalable deep learning search applications in minutes.

Neural Search Example

Query: a happy potato

Query vector: [0.01, 0.07, 0.35,]

Result vectors: [[0.01, 0.06, 0.34,]
[0.02, 0.06, 0.38, ..]

...



Preliminaries

02

Fundamental

3D Data format

- 3D data is acquired by various types of 3D scanners, LiDARs, and RGB-D cameras
- Usually represented in different formats

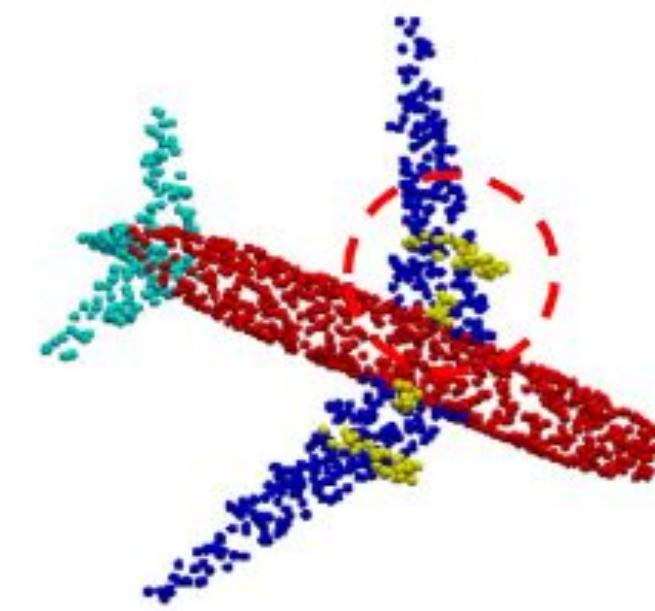


(a) Real-image

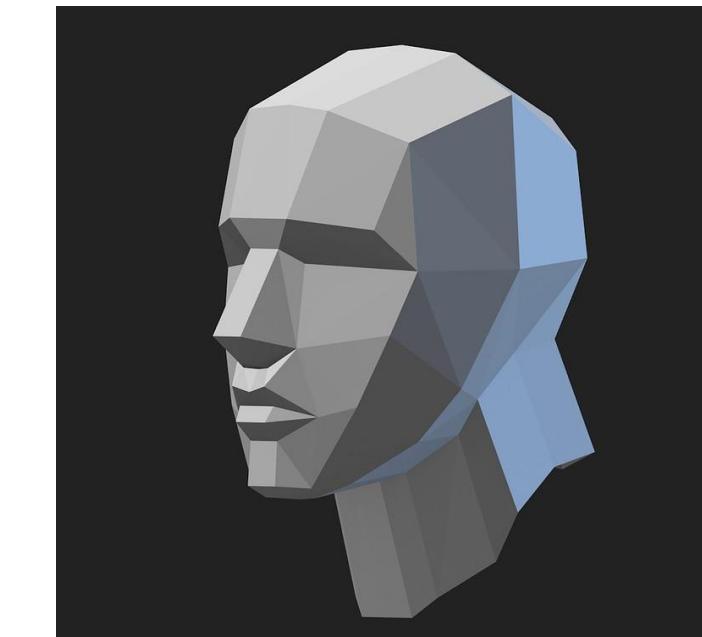


(b) Depth-map

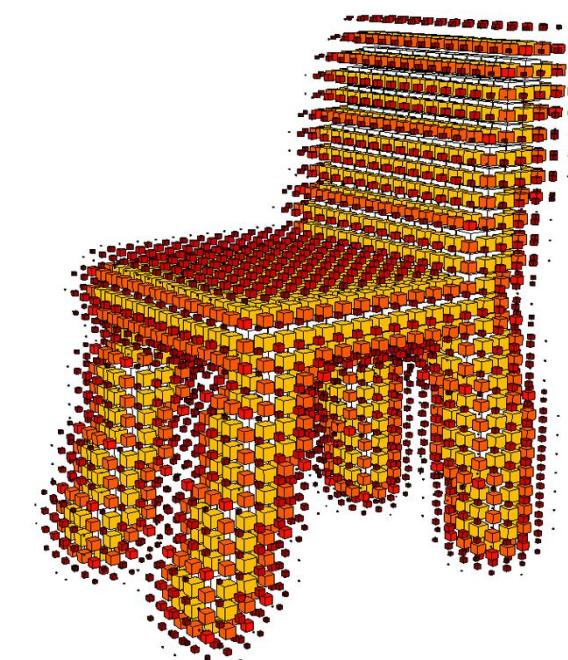
depth map



point cloud



mesh



volumetric grids

Preliminaries

FBX & glTF Data

FBX is a **proprietary** file format developed by Kaydara and owned by Autodesk since 2006. It is used to provide interoperability between digital content creation applications. FBX is also part of Autodesk Gameware, a series of video game middleware.

glTF (**opensource**) is a standard file format for three-dimensional scenes and models. A glTF file uses one of two possible file extensions, .gltf or .glb. A .gltf file may be self-contained or may reference external binary and texture resources, while a .glb file is entirely self-contained.



☰ README.md

FBX2gltf

License BSD 3-Clause

This is a command line tool for converting 3D model assets on Autodesk's venerable [FBX](#) format to [glTF 2.0](#), a modern runtime asset delivery format.

Precompiled binaries releases for Windows, Mac OS X and Linux may be found [here](#).

Bleeding-edge binaries for Windows may be found [here](#). Linux and Mac OS X to come; meanwhile, you can [build your own](#).

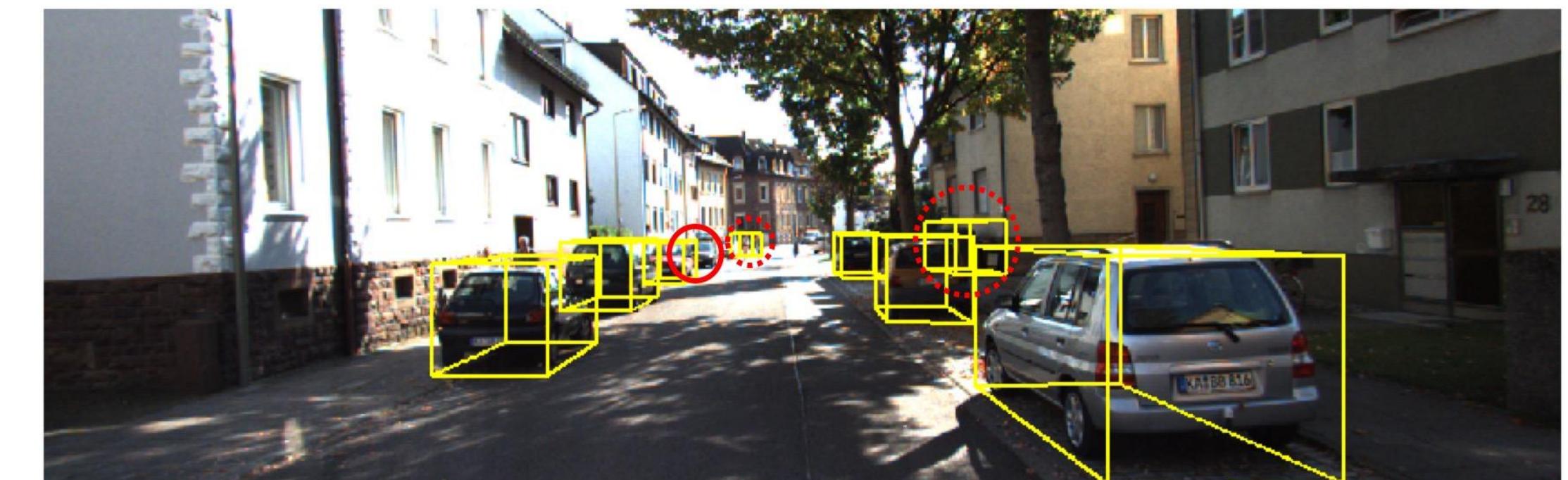
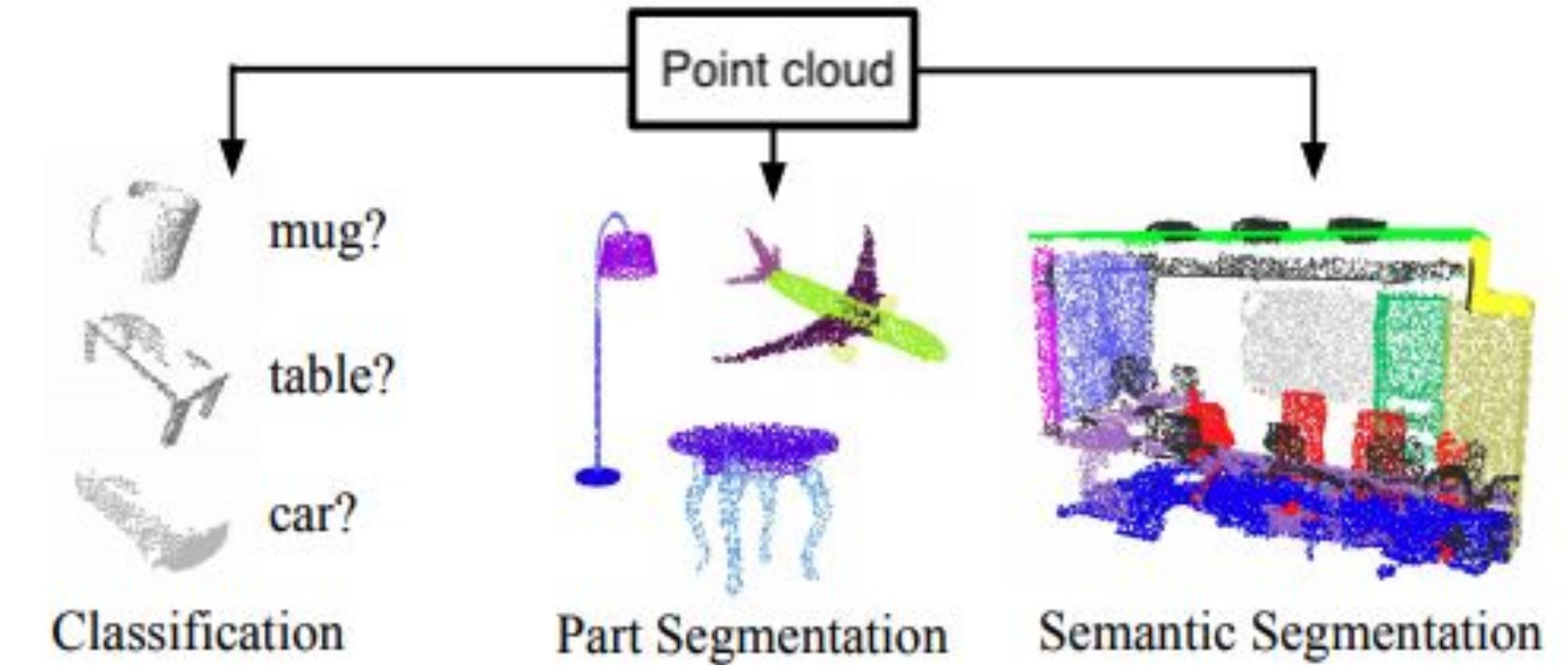
[build unknown](#) [build passing](#)

<https://github.com/facebookincubator/FBX2gltf>

Problem Formulation

How to use 3D data?

- Shape classification
- Segmentation
- Object detection



object detection

Recent Work

Model	Time
PointNet	2017
PointNet++	2017
PointCNN	2018
PVCNN	2019
VoteNet	2019
PointConv	2020
PointCLIP	2021
PAConv	2021
FastPointTransformer	2022
ModelNet-C	2022
.....

Problem Formulation

03

Problem Formulation

Search, Recommendation

Mesh: the basic query & index element

Search

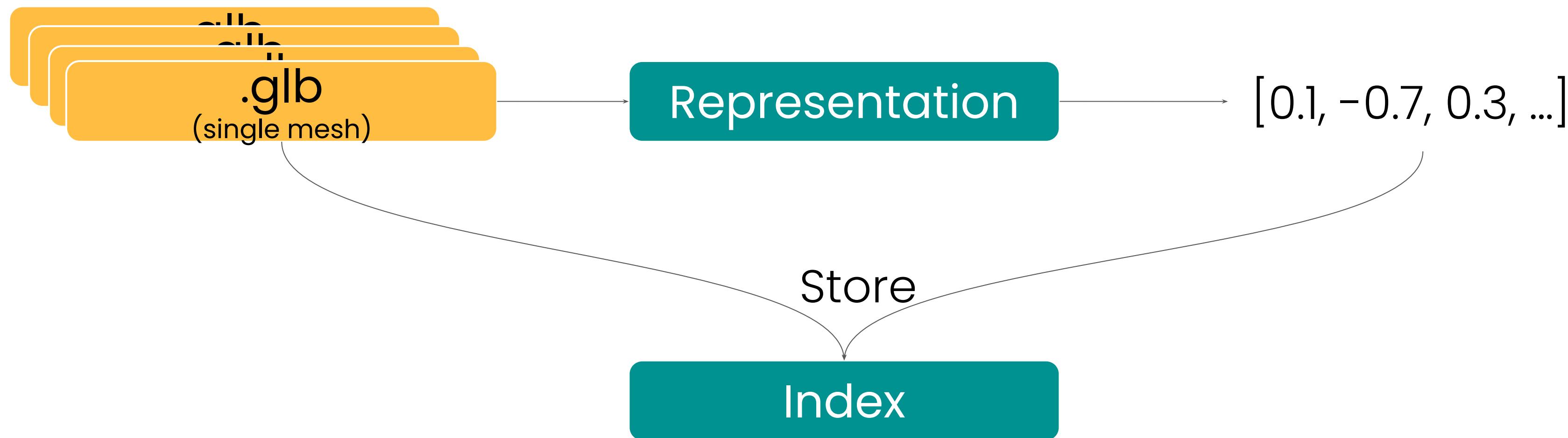
Given an *unobserved* mesh, find top-K visually similar neighbours from all indexed data.

Recommendation

Given a *indexed* mesh, find top-K visually similar neighbours from all indexed data.

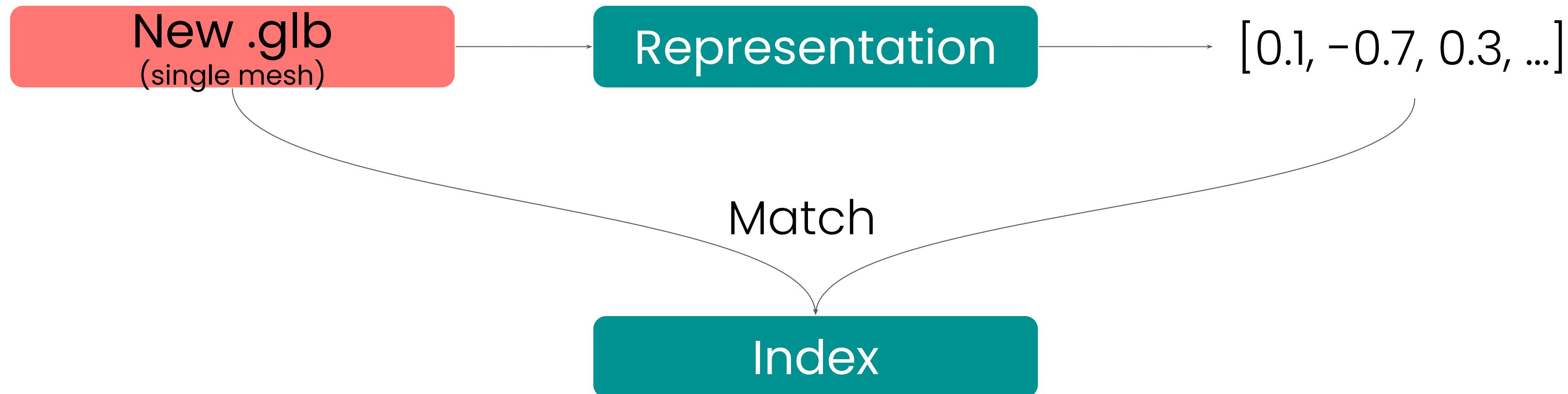
Problem Formulation

Search: Index



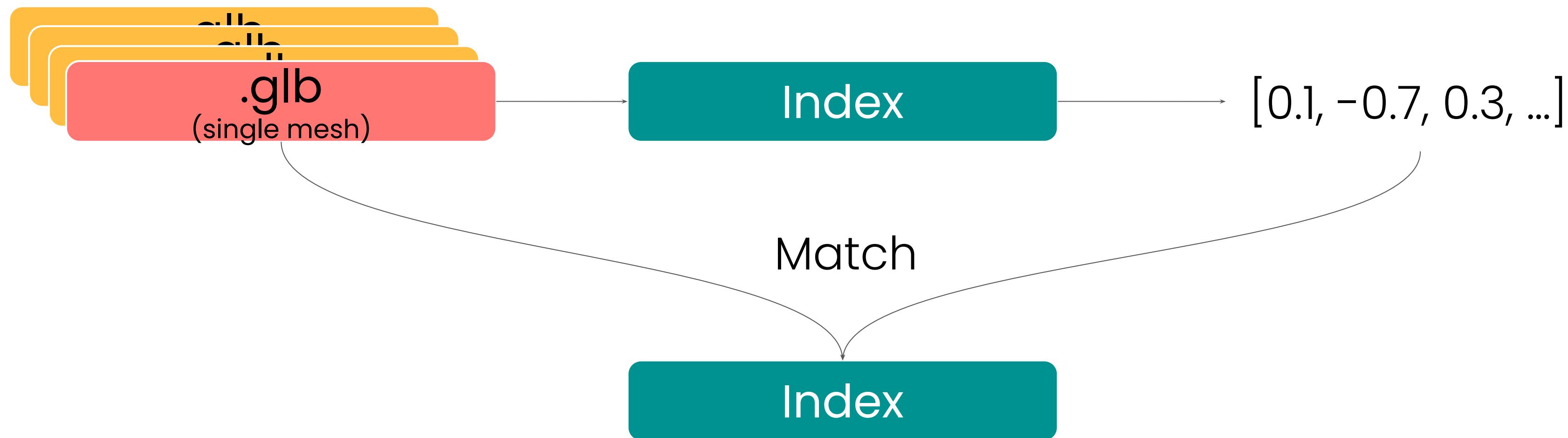
Problem Formulation

Search: Query



Problem Formulation

Recommendation



Solution Engineering

Representation Learning

Mesh-based:

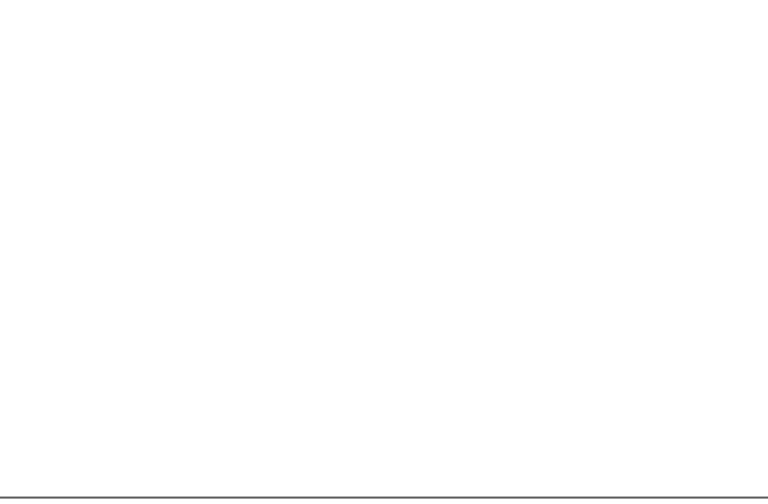
- Geometry properties



[0.1, -0.7, 0.5,]

Point Cloud-based:

View Image-based



[0.9, 0.2, 0.3,]

Representation Learning

Mesh-based Geometric Properties



Fixed-length properties

Area: 0.00030684633550101244
BodyCount: 378
Bounds: [[-2.97899009e-03 6.91652303e-06 -1.44493487e-03], [2.97899009e-03 1.86841190e-02 1.61421520e-03]]
CenterOfMass: [7.12339424e-12 7.39450446e-03 -6.80447296e-05]
Centroid: [2.14374024e-11 9.49250934e-03 -4.37685498e-05]
EulerNumber: 378
Mass: 1.1205215085566987e-07
Scale: 0.01984171914811901
Volume: 1.1205215085566987e-07

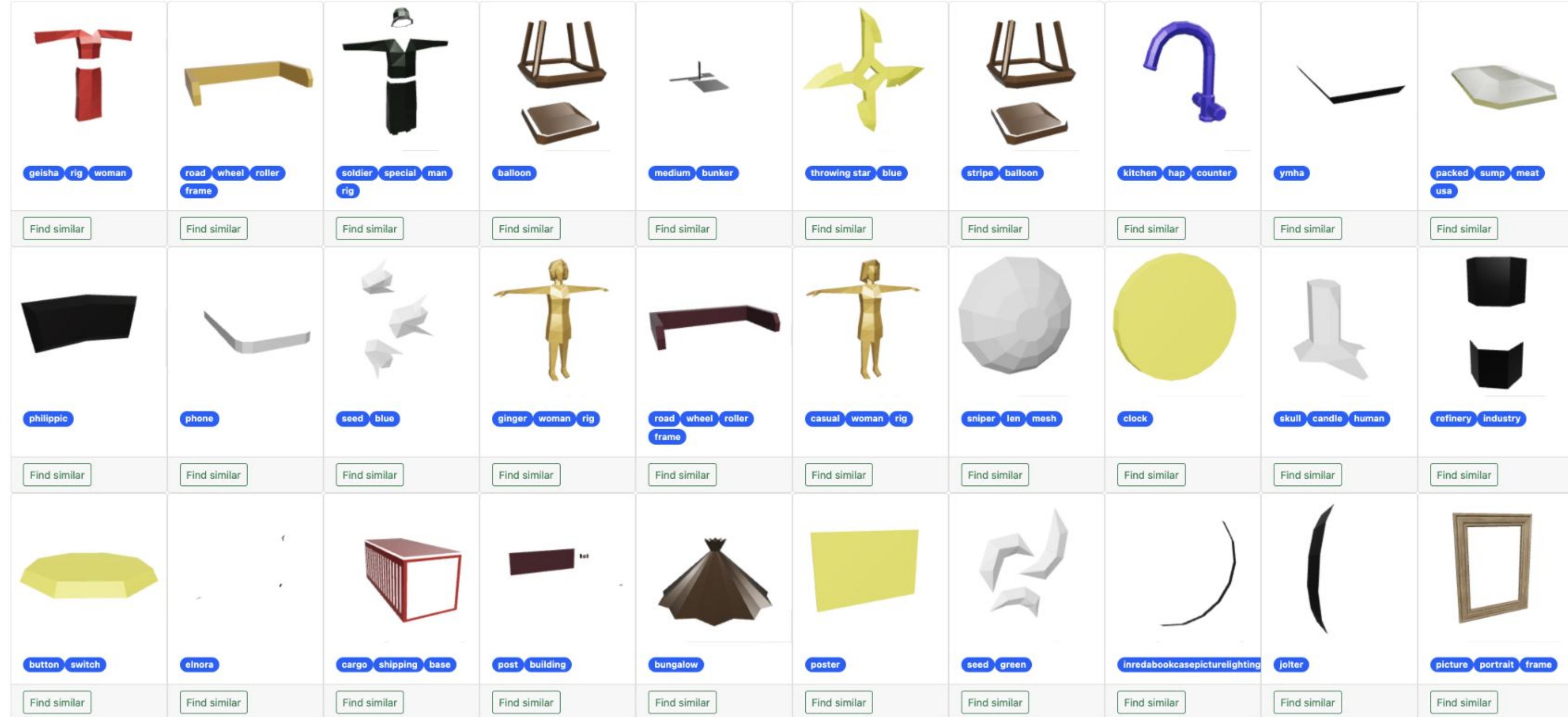
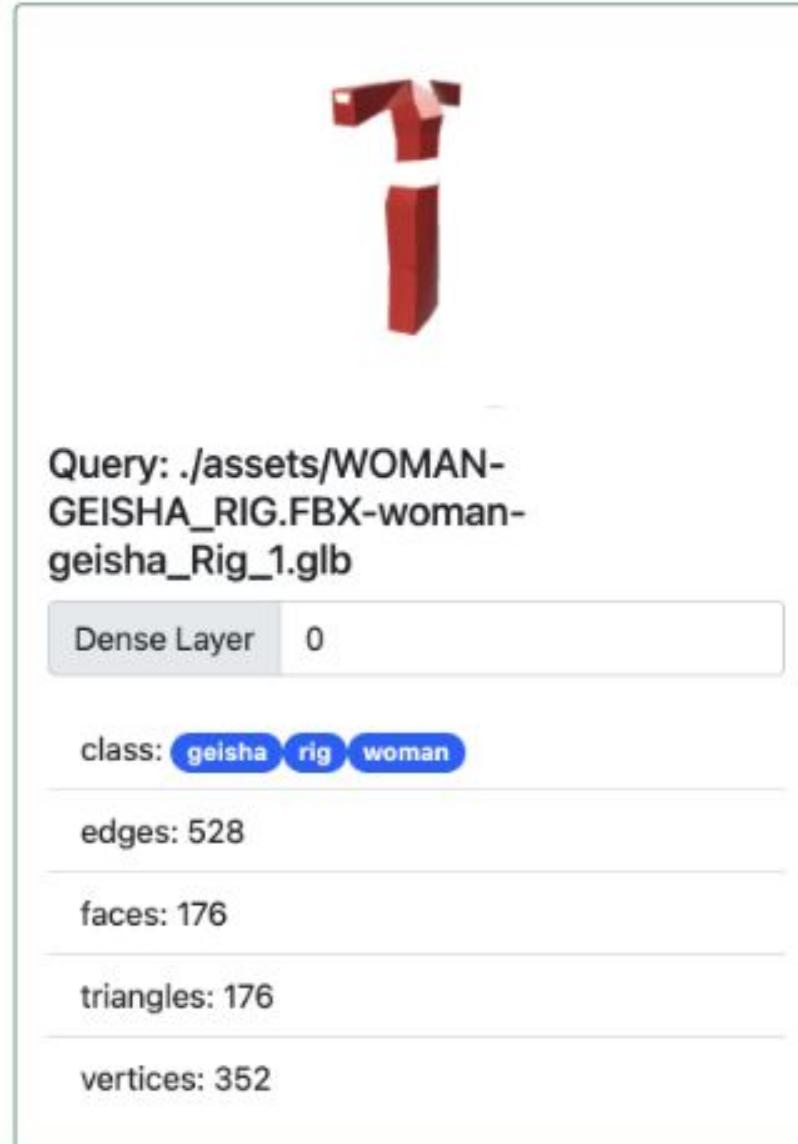


Fixed-length properties

Area: 18.820192049265003
BodyCount: 58
Bounds: [[-1.057832 0.842691 -0.92879516], [1.057832 2.99272704 0.80760008]]
CenterOfMass: [0.00218869 1.07203907 0.01342663]
Centroid: [0.00833604 1.63144851 -0.03551894]
EulerNumber: 57
Mass: 2.0362587087359363
Scale: 3.4804823763958166
Volume: 2.0362587087359363



Most similar 30



Representation Learning

No-go: Mesh-based Geometric Properties

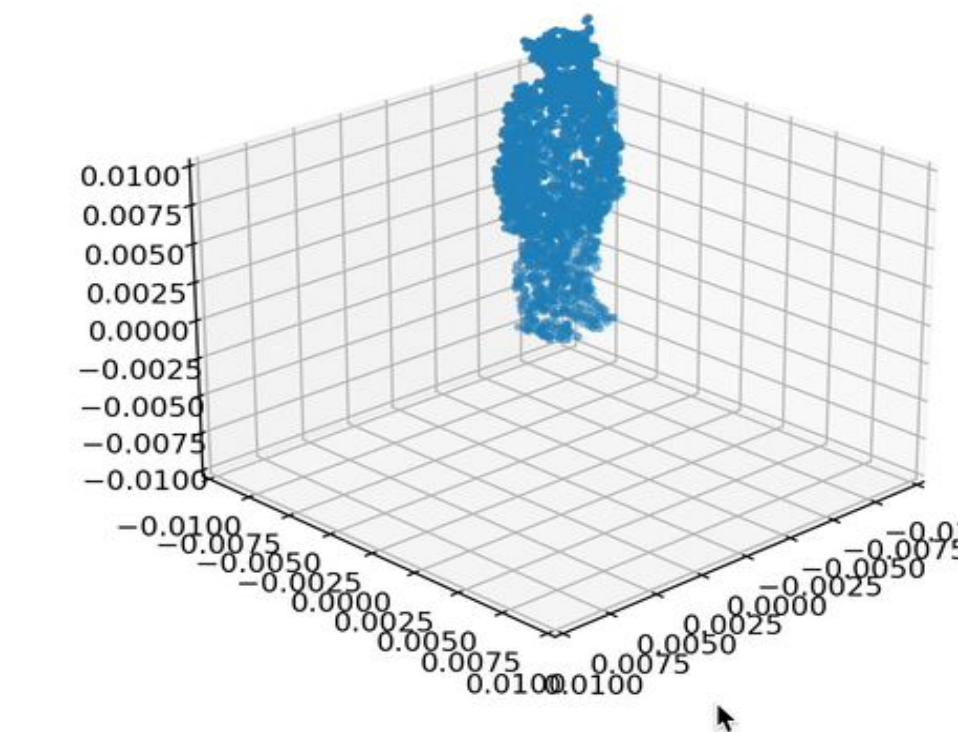
- ✓ Maintain *certain* invariances of the transformation
- ✓ Efficient encoding
- 😔 Heuristic & handcrafted
- 😔 Depending on the mesh quality (number of vertices, faces, edges)
- 😔 Can be used as unique hash but not similarity hash

Representation Learning

Point Cloud-based



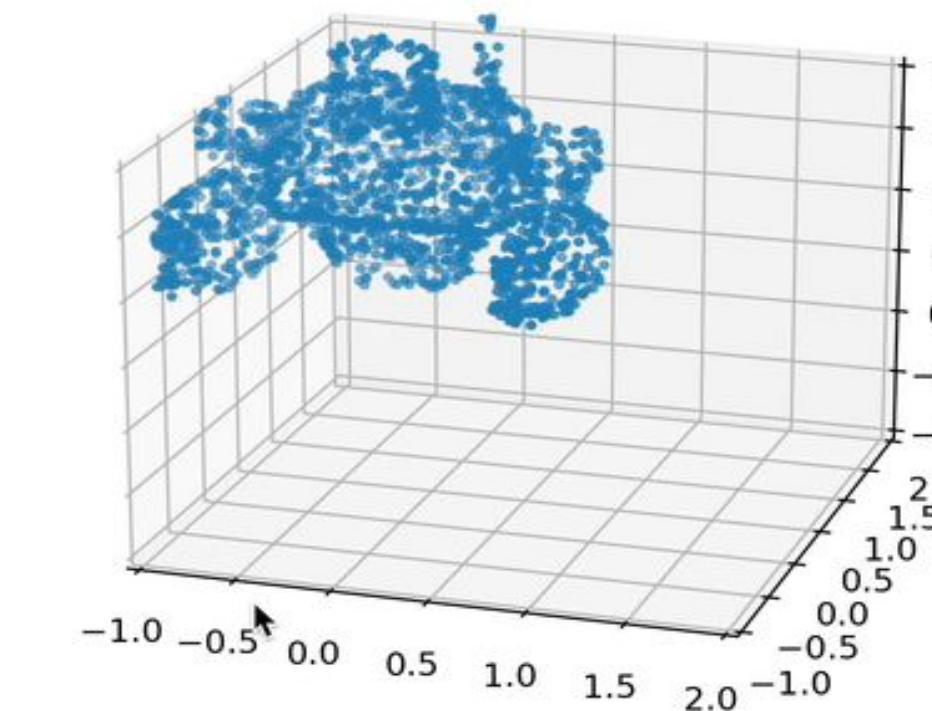
Sampling 2048 points



2048 x 3
matrix



Sampling 2048 points



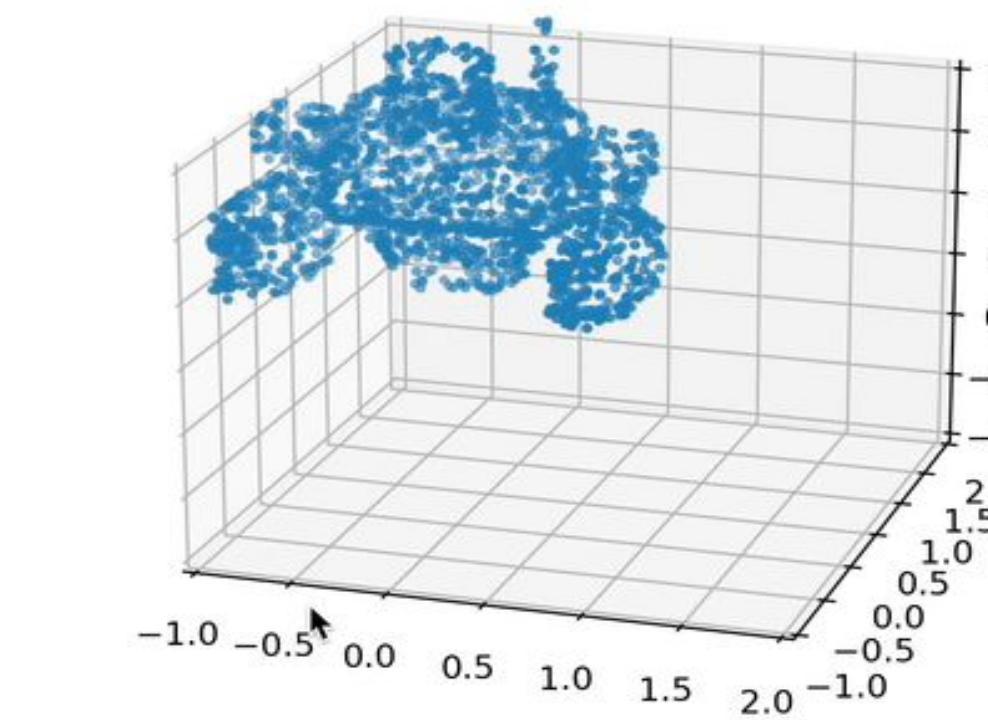
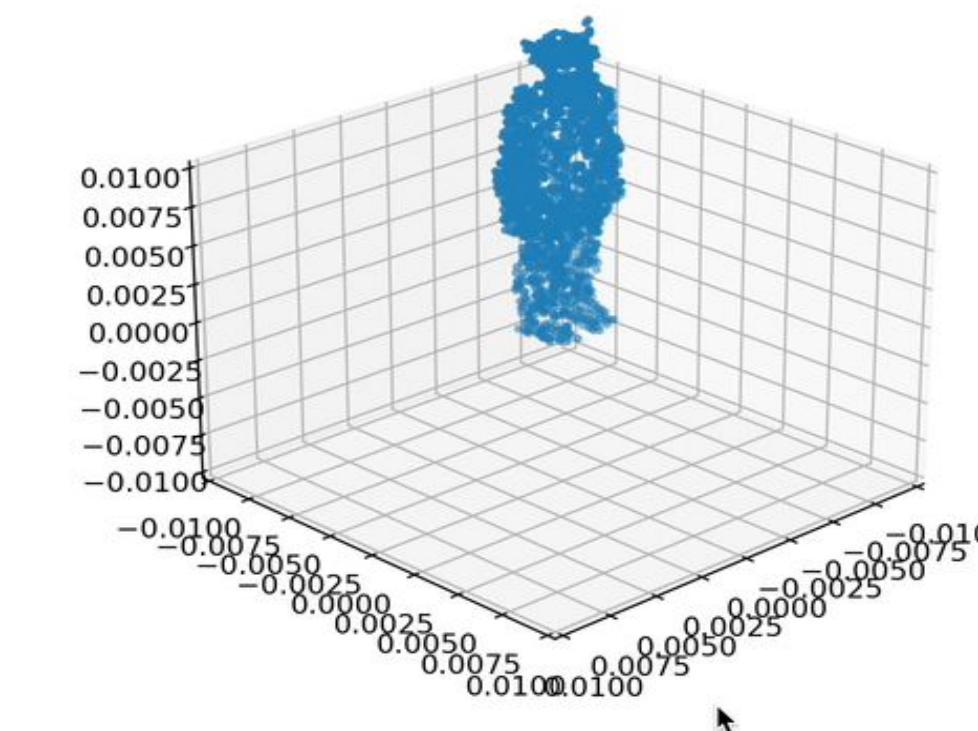
2048 x 3
matrix

Representation Learning

Point Cloud-based



Sampling 2048 points



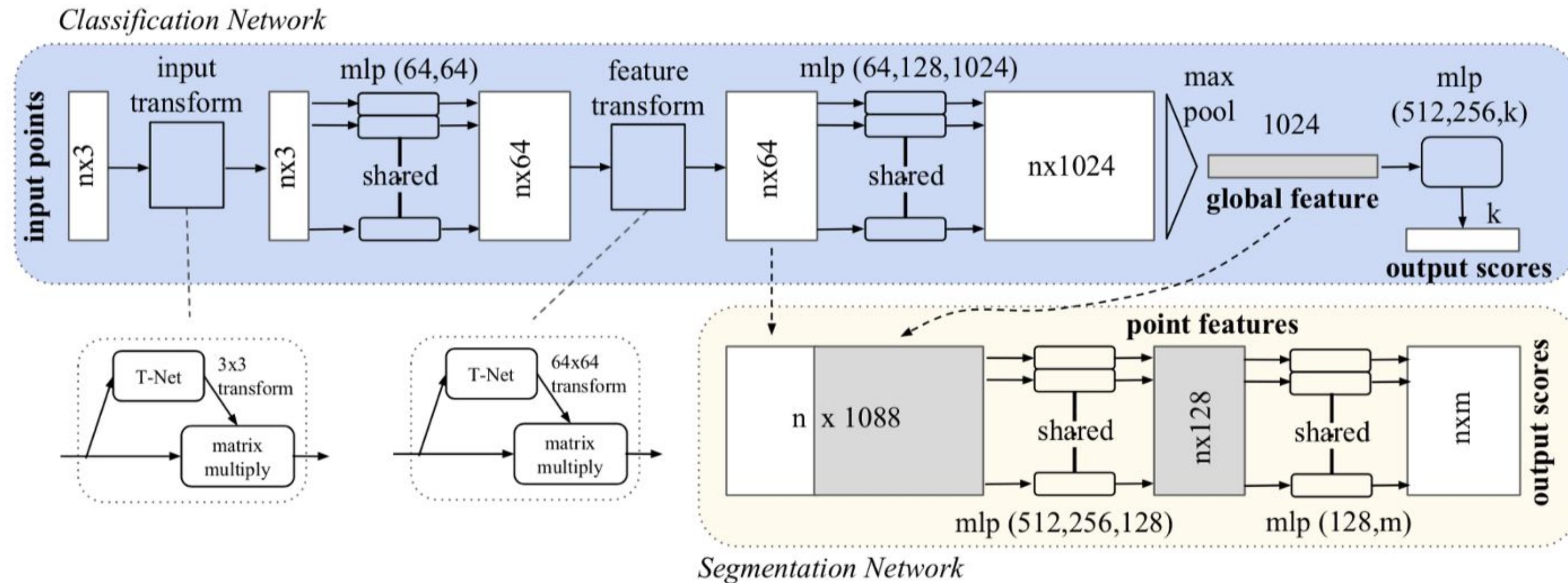
2048 x 3
matrix

**Order
relevant**

2048 x 3
matrix

Representation Learning

PointNet

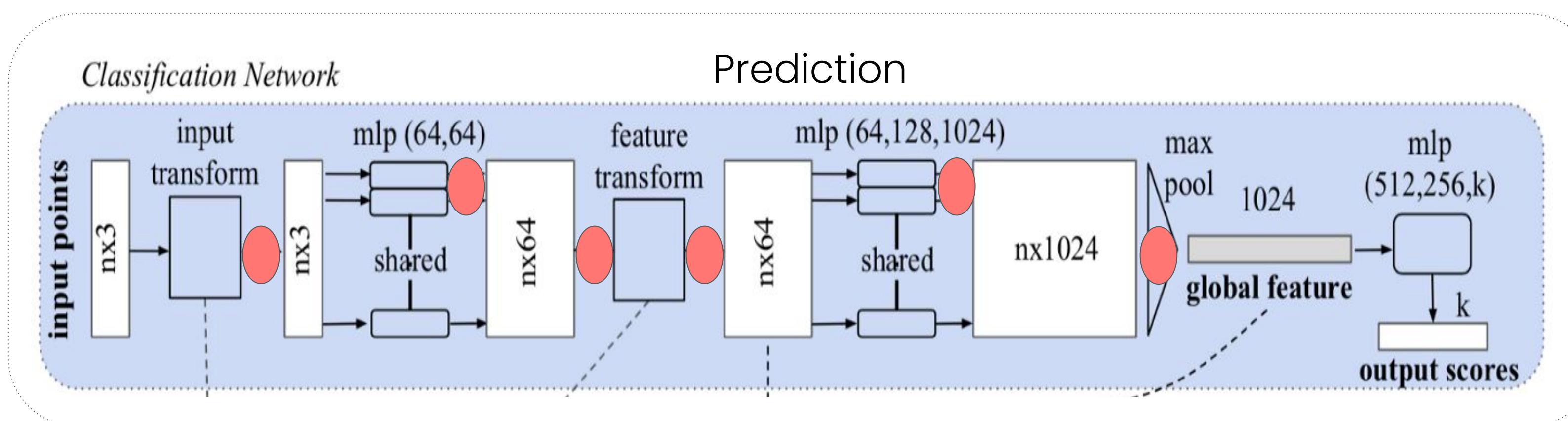
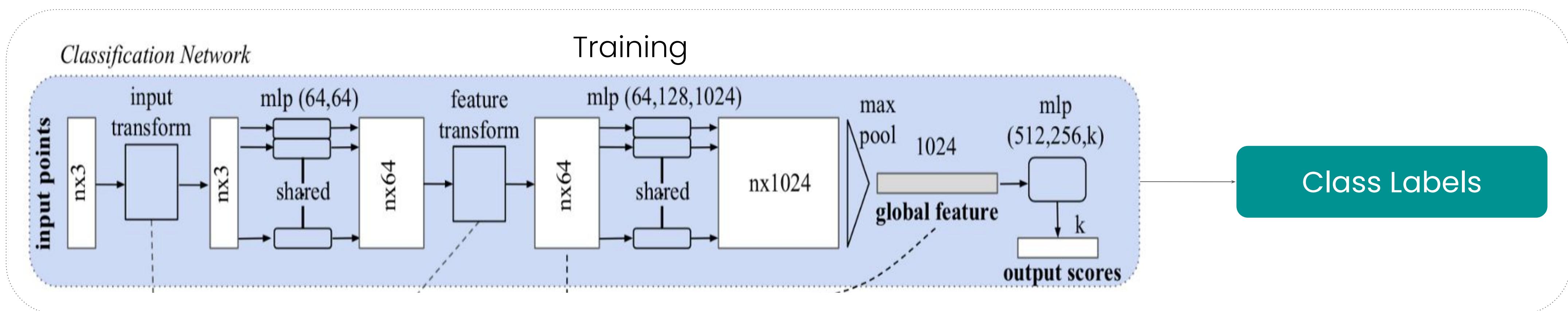


PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

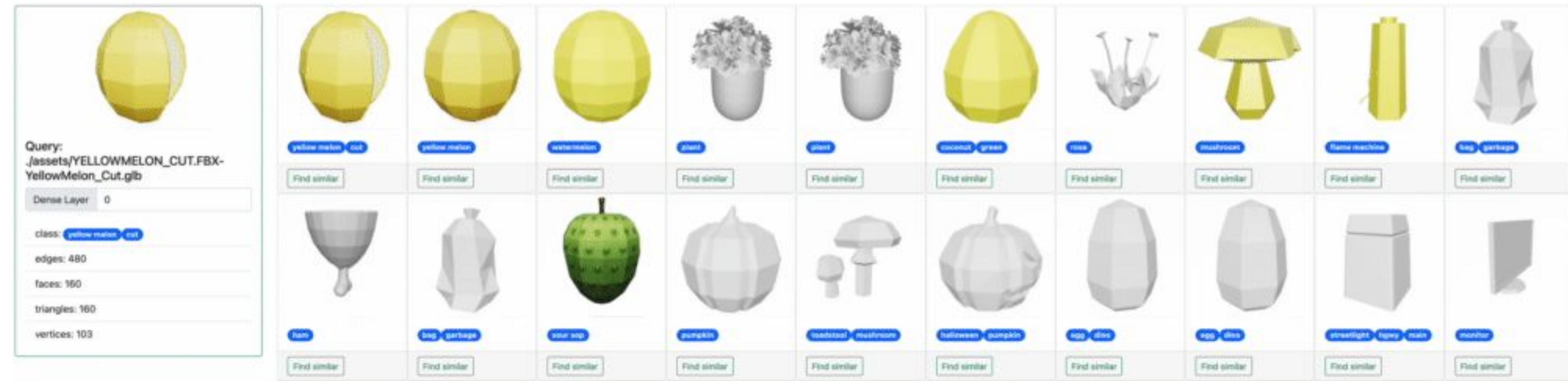
<https://arxiv.org/pdf/1612.00593.pdf>

Representation Learning

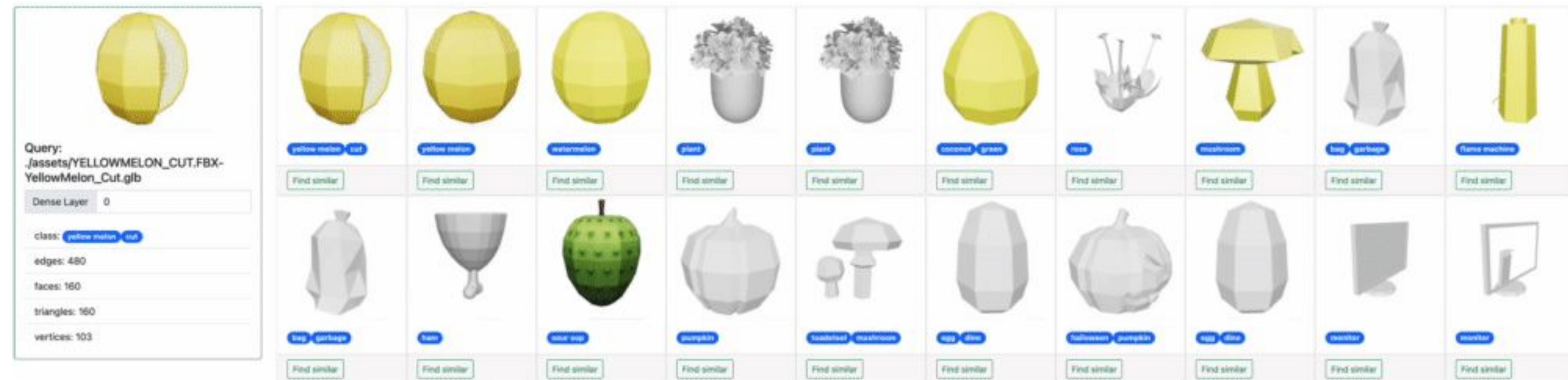
PointNet



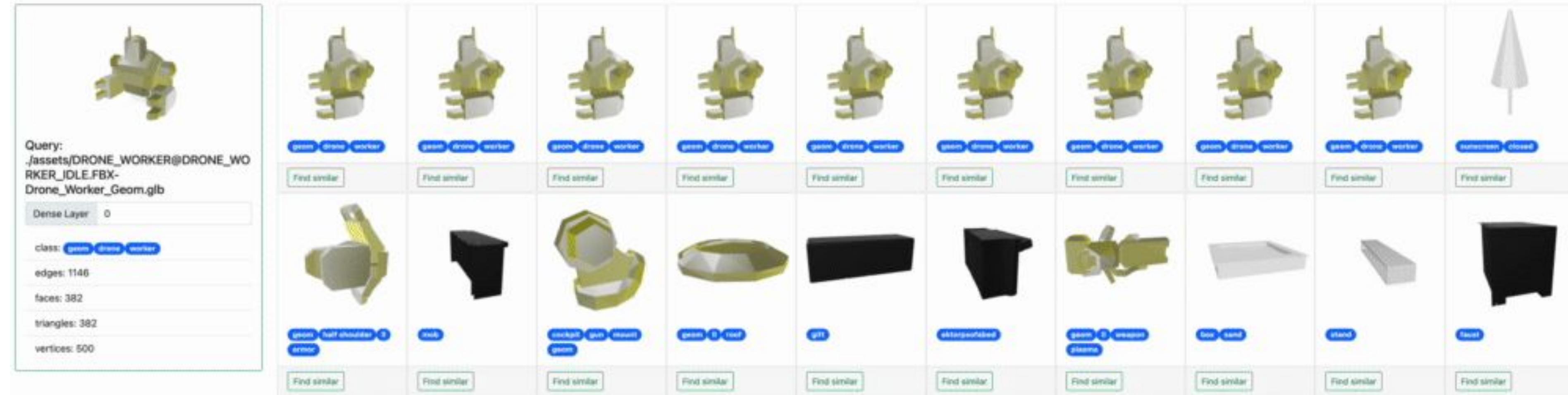
Geo-properties



PointNet



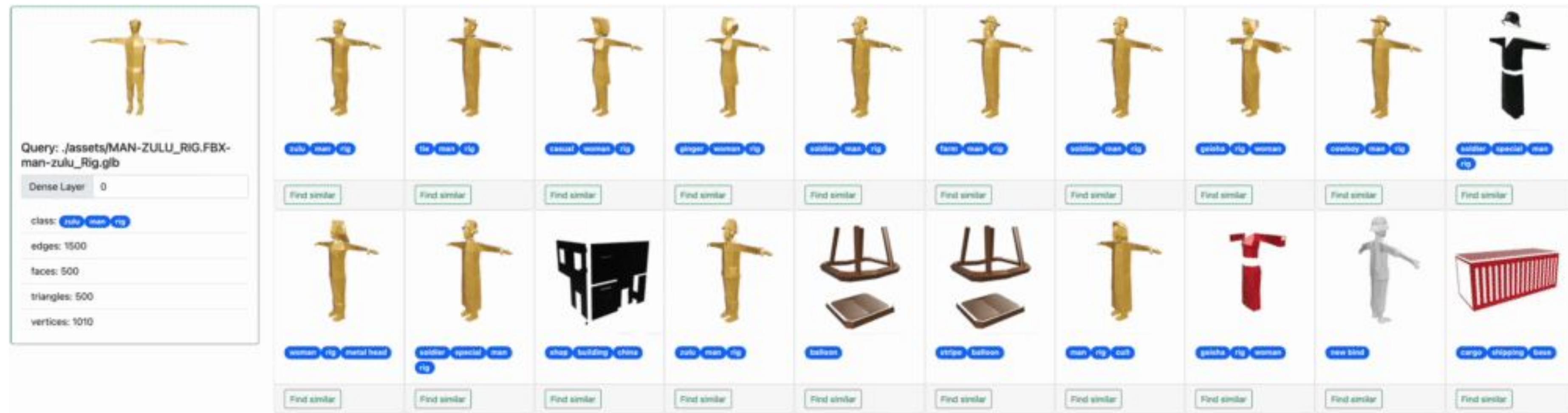
Geo-properties



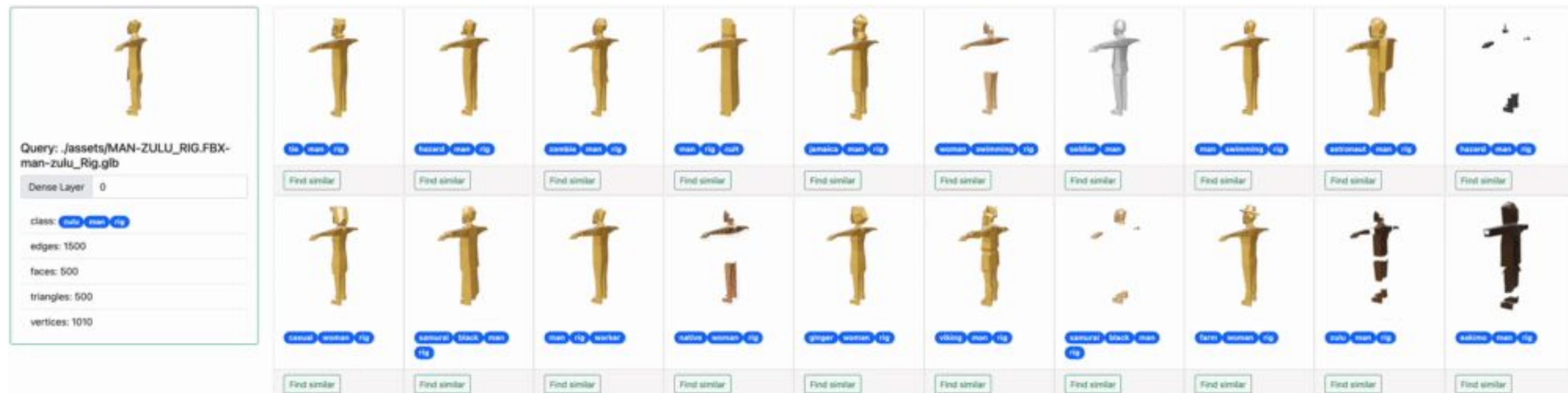
PointNet



Geo-properties



PointNet



Advanced: Cross-modality



CLIP

- CLIP (Contrastive Language-Image Pre-Training)
- Trained to match images with their corresponding natural language description

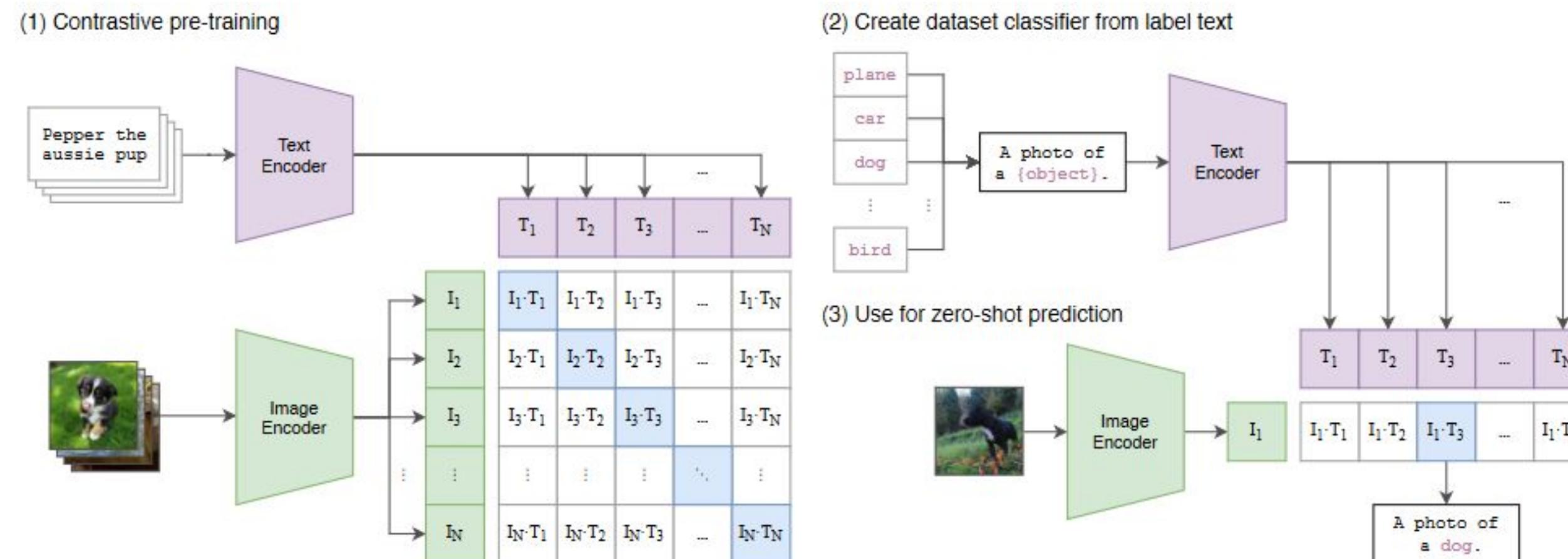
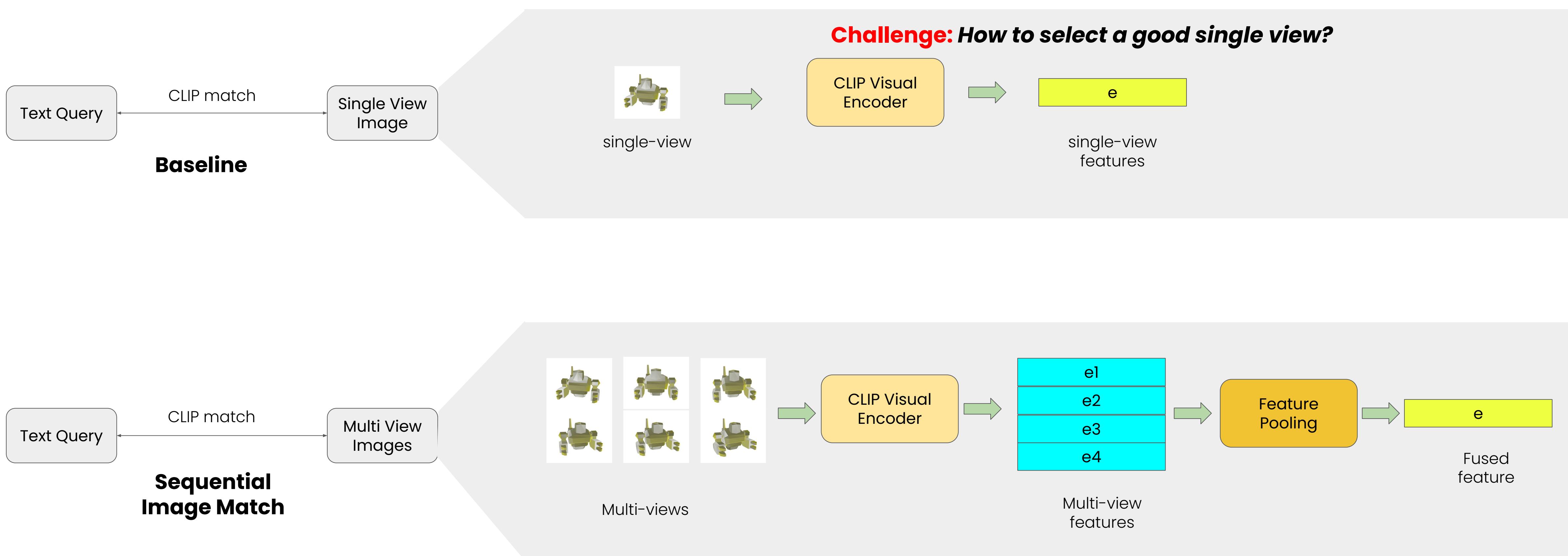


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Multi-view based approach

CLIP Match



★ **Sequential Image Match:** is a general/popular matching problem in practice, and its solutions can be easily applied to video search

Examples in Jina World

04

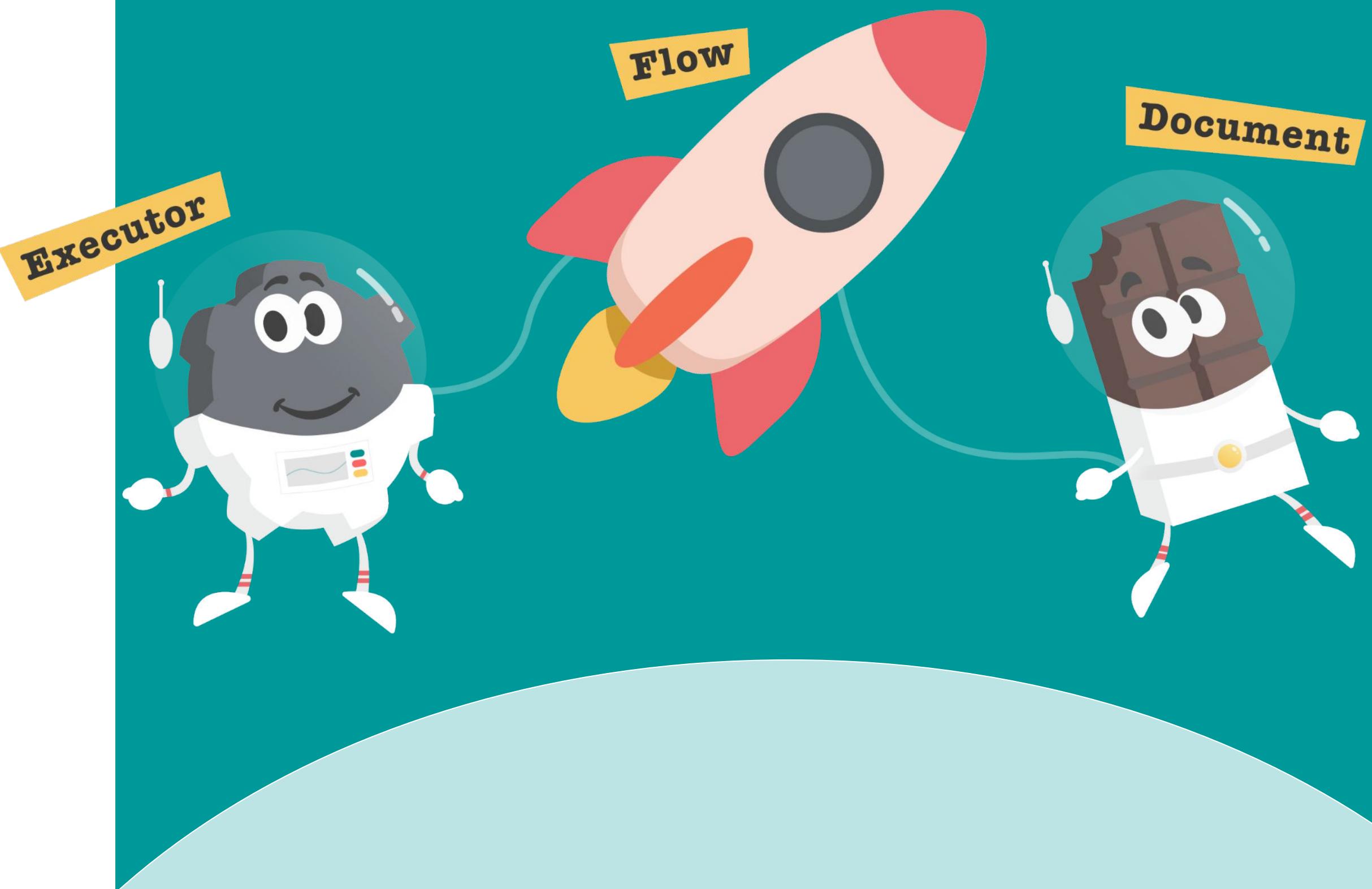
Fundamental

Document, Executor & Flow

Document is the basic data structure in Jina;

Executor is a group of functions that takes Documents as IO.

Flow is how Jina streamlines and distributes Executors.



一起点亮

GitHub

欢迎大家 Star





Fundamental

Hello, world



一起点亮

GitHub

欢迎大家 Star



```
1 #!pip install jina
2 from jina import DocumentArray, Executor, Flow, requests
3
4 class MyExec(Executor):
5
6     @requests
7     async def foo(self, docs: DocumentArray, **kwargs):
8         for d in docs:
9             d.text += 'hello, world!'
10
11 f = Flow().add(uses=MyExec).add(uses=MyExec)
12
13 with f:
14     r = f.post('/', DocumentArray.empty(2))
15     print(r.texts)
```



Fundamental

Imports



一起点亮

GitHub

欢迎大家 Star



```
1 #!pip install jina
2 from jina import DocumentArray, Executor, Flow, requests
3
4 class MyExec(Executor):
5
6     @requests
7     async def foo(self, docs: DocumentArray, **kwargs):
8         for d in docs:
9             d.text += 'hello, world!'
10
11 f = Flow().add(uses=MyExec).add(uses=MyExec)
12
13 with f:
14     r = f.post('/', DocumentArray.empty(2))
15     print(r.texts)
```



Fundamental

Define Executor



一起点亮

GitHub

欢迎大家 Star



```
1 #!pip install jina
2 from jina import DocumentArray, Executor, Flow, requests
3
4 class MyExec(Executor):
5
6     @requests
7     async def foo(self, docs: DocumentArray, **kwargs):
8         for d in docs:
9             d.text += 'hello, world!'
10
11 f = Flow().add(uses=MyExec).add(uses=MyExec)
12
13 with f:
14     r = f.post('/', DocumentArray.empty(2))
15     print(r.texts)
```



Fundamental

Build Flow



一起点亮

GitHub

欢迎大家 Star



```
1 #!pip install jina
2 from jina import DocumentArray, Executor, Flow, requests
3
4 class MyExec(Executor):
5
6     @requests
7     async def foo(self, docs: DocumentArray, **kwargs):
8         for d in docs:
9             d.text += 'hello, world!'
10
11 f = Flow().add(uses=MyExec).add(uses=MyExec)
12
13 with f:
14     r = f.post('/', DocumentArray.empty(2))
15     print(r.texts)
```



Fundamental

Send/Recv Data



一起点亮
GitHub

欢迎大家 Star



```
1 #!pip install jina
2 from jina import DocumentArray, Executor, Flow, requests
3
4 class MyExec(Executor):
5
6     @requests
7     async def foo(self, docs: DocumentArray, **kwargs):
8         for d in docs:
9             d.text += 'hello, world!'
10
11 f = Flow().add(uses=MyExec).add(uses=MyExec)
12
13 with f:
14     r = f.post('/', DocumentArray.empty(2))
15     print(r.texts)
```



Fundamental

Run it!

```
➜ jina git:(master) x
```



一起点亮

GitHub

欢迎大家 Star





3D Mesh in Jina

3D Mesh Document

What should be included to represent the 3D mesh data?

1. Point-clouds

```
from docarray import Document

doc = Document(uri='drone.glb')

doc.load_uri_to_point_cloud_tensor(2048)
```

2. Multi-view images: captured by cameras with different angles

```
# single view
doc.uri = f'drone.jpg'

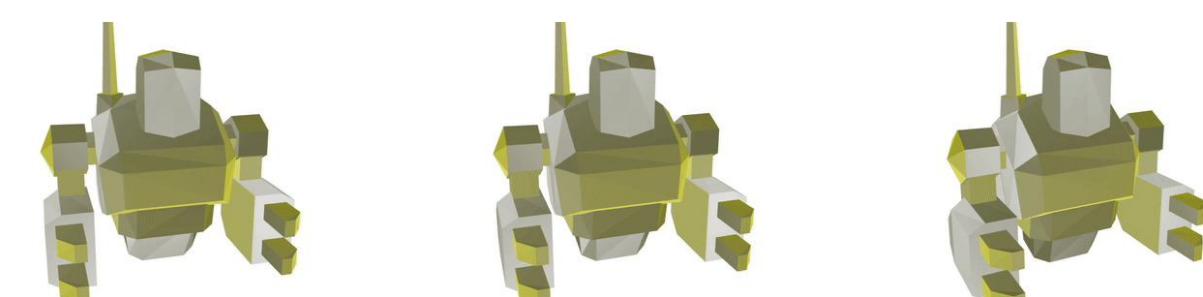
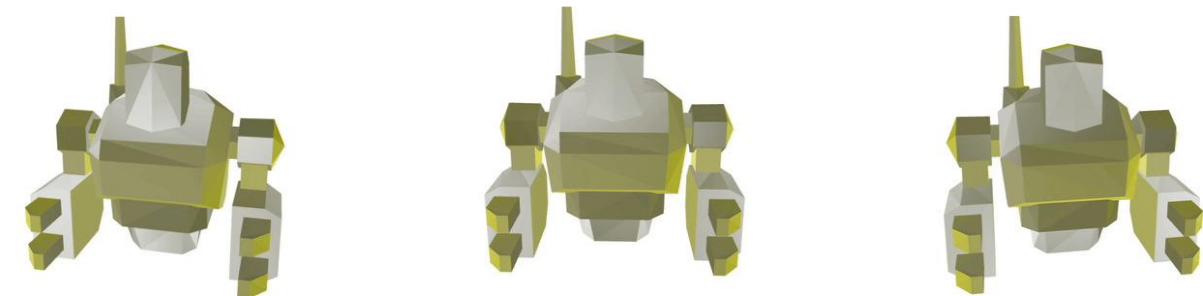
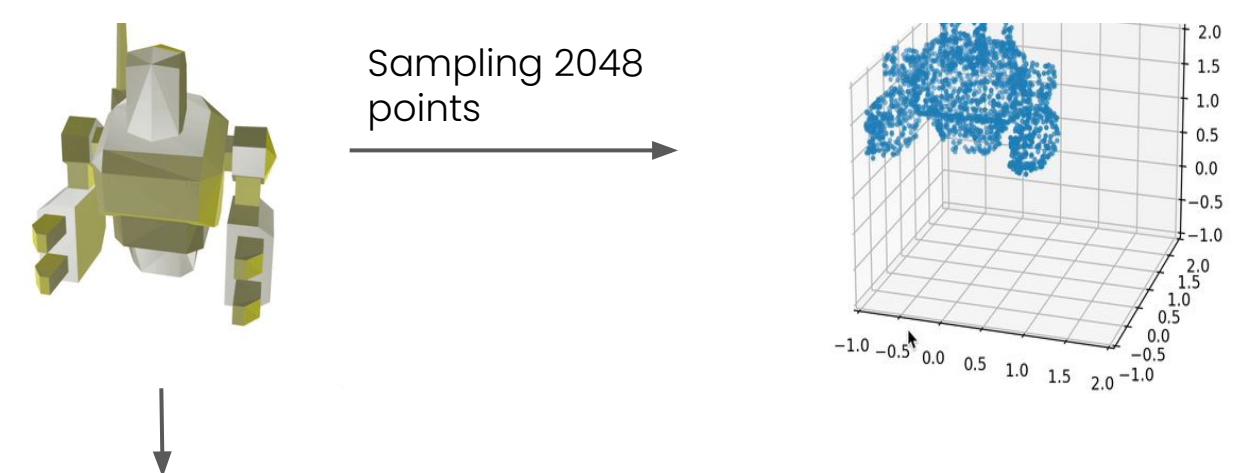
# 6 different views
for k in range(6):
    doc.chunks.append(Document(uri=f'drone_{k}.jpg'))
```

DocArray is a library for nested, unstructured, multimodal data in transit, including text, image, audio, video, 3D mesh, etc. It allows deep-learning engineers to efficiently process, embed, search, recommend, store, and transfer the multi-modal data with a Pythonic API.

Door to cross-/multi-modal world: super-expressive data structure for representing complicated/mixed/nested text, image, video, audio, 3D mesh data. The foundation data structure of [Jina](#), [CLIP-as-service](#), [DALL-E Flow](#), [DiscoArt](#) etc.

Pythonic experience: designed to be as easy as a Python list. If you know how to Python, you know how to DocArray. Intuitive idioms and type annotation simplify the code you write.

Data science powerhouse: greatly accelerate data scientists' work on embedding, k-NN matching, querying, visualizing, evaluating via Torch/TensorFlow/ONNX/PaddlePaddle on CPU/GPU.



一起点亮
GitHub
欢迎大家 Star





Examples

Single-modality

Hub

Explore Docs Create [Login](#) [Sign up](#)

jina-ai / TrimeshLoader
Loads triangular meshes and extract point cloud
latest · Public · Published 3 months ago

[loader](#) [mesh](#) [3d](#) [trimesh](#)

Try in Sandbox [Text](#)

[Send](#)

Customized tags [Create](#)

Generate application code

Dev Environment [In Development \(Python\)](#) [In Production](#)
Entrypoint [In DocArray](#) [In Jina](#)
Executor Usage [Via Sandbox](#) [Via Docker](#) [Via Source](#)
Tag [latest](#)
Code Snippet [View](#)

```
from docarray import Document, DocumentArray
da = DocumentArray([Document(text='hello')])
r = da.post('jinahub+sandbox://TrimeshLoader/latest')
```

jina-ai / executor-3d-encoder [Public](#)

Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

main 1 branch 2 tags Go to file Add file Code About

ZiniuYu docs: correct executor name in hub link 2a9df04 on Jun 17 46 commits
.github/workflows ci: push main also trigger jina hub push 5 months ago
executor fix: add test 3 months ago
tests fix: add test 3 months ago
citianpore fix: ignore checkpoints 7 months ago

flow.yml

```
jtype: Flow
with:
  protocol: grpc
  port: 12345
```

executors:
 - name: loader
 uses: 'jinahub://TrimeshLoader'
 install_requirements: true

 - name: encoder
 uses: 'jinahub://MeshDataEncoder'
 timeout_ready: -1
 install_requirements: true
 uses_with:
 pretrained_model: PointConv-Shapenet-d512

 - name: indexer
 uses: 'jinahub://AnnLiteIndexer/latest'
 uses_with:
 dim: 512
 limit: 10

Releases 2 patch v0.3.0 (Latest) on May 6

Try release Packages packages published Publish your first package

一起点亮 GitHub 欢迎大家 Star



Examples

Cross modality

Search 3D assets by keywords or phrases e.g. 路标, 跑酷 and 楼梯, etc.

万圣节

Search



万圣屋 / 粉福

```
{'cosine': {'value': 0.52678317, 'opName': '', 'description': '', 'operands': [], 'refId': ''}}
```



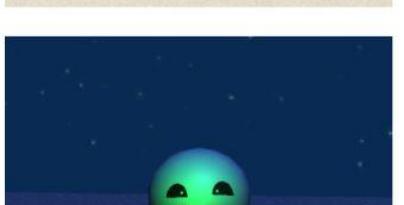
今年南瓜

```
{'cosine': {'value': 0.5283447, 'opName': '', 'description': '', 'operands': [], 'refId': ''}}
```



万圣节跑酷素材

```
{'cosine': {'value': 0.5305282, 'opName': '', 'description': '', 'operands': [], 'refId': ''}}
```



幽灵

```
{'cosine': {'value': 0.5307214, 'opName': '', 'description': '', 'operands': [], 'refId': ''}}
```



火速给我上个热门.南瓜耶

```
{'cosine': {'value': 0.5326485, 'opName': '', 'description': '', 'operands': [], 'refId': ''}}
```



flow.yml

```

1 jtype: Flow
2 with:
3   protocol: grpc
4   port: 12345
5   executors:
6     - name: encoder
7       uses: 'jinahub://CLIP TorchEncoder/latest'
8       timeout_ready: -1
9       uses_with:
10         name: M-CLIP/XLM_Roberta-Large-Vit-B-32
11
12     - name: indexer
13       uses: 'jinahub://AnnLiteIndexer/latest'
14       uses_with:
15         dim: 512
16         limit: 10

```



一起点亮

GitHub

欢迎大家 Star





clip-as-service



CLIP-as-service

Embedding image and sentence
into fixed-length vectors via CLIP

<https://github.com/jina-ai/clip-as-service> 

<https://clip-as-service.jina.ai/playground/reasoning/>



 [jina-ai / clip-as-service](#) Public

 Fork 2k  Starred 10.6k

 README.md


CLIP-as-service

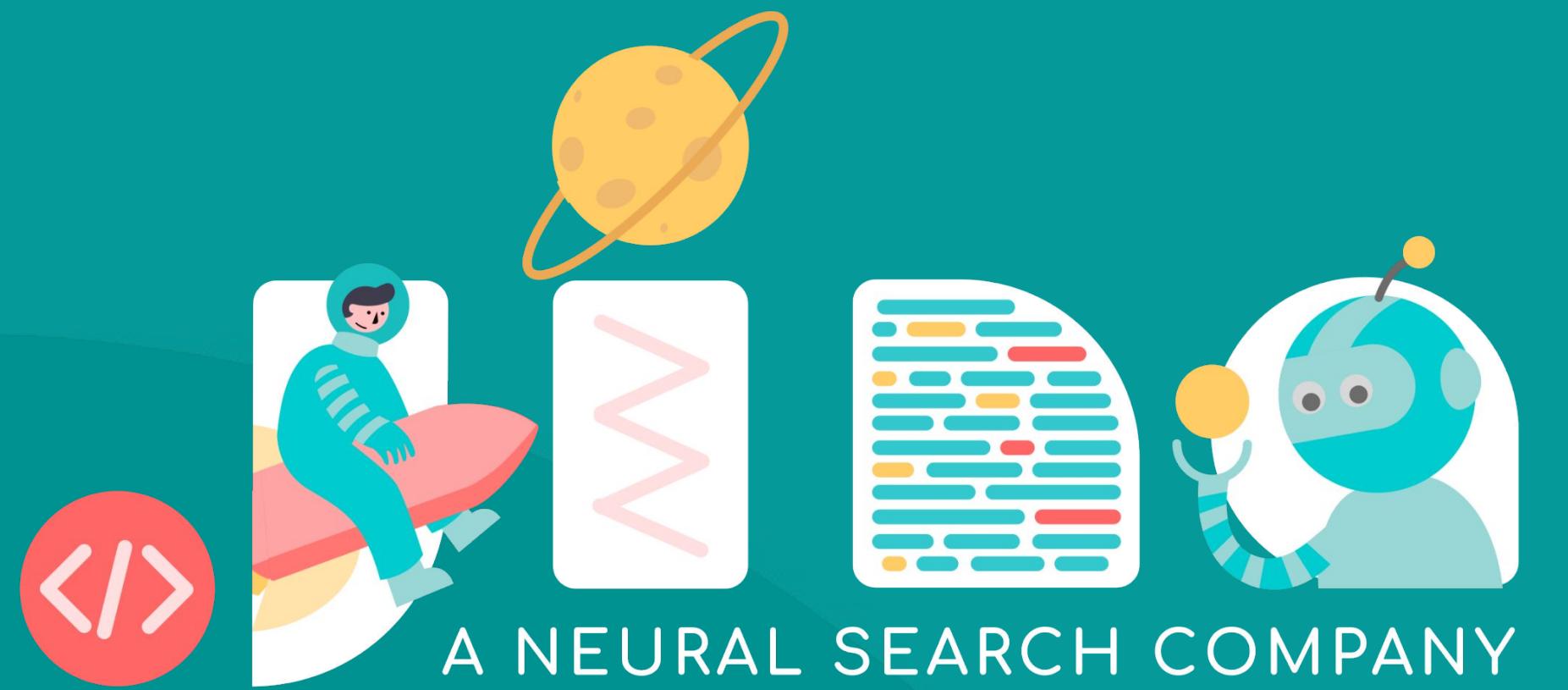
Embed images and sentences into fixed-length vectors with CLIP

Release v0.5.1 Slack 3.1k coverage 84% Host on Google Colab (GPU/TPU)

CLIP-as-service is a low-latency high-scalability service for embedding images and text. It can be easily integrated as a microservice into neural search solutions.

-  **Fast:** Serve CLIP models with TensorRT, ONNX runtime and PyTorch w/o JIT with 800QPS^[*]. Non-blocking duplex streaming on requests and responses, designed for large data and long-running tasks.
-  **Elastic:** Horizontally scale up and down multiple CLIP models on single GPU, with automatic load balancing.
-  **Easy-to-use:** No learning curve, minimalist design on client and server. Intuitive and consistent API for image and sentence embedding.
-  **Modern:** Async client support. Easily switch between gRPC, HTTP, WebSocket protocols with TLS and compression.
-  **Integration:** Smooth integration with neural search ecosystem including [Jina](#) and [DocArray](#). Build cross-modal and multi-modal solutions in no time.

[*] with default config (single replica, PyTorch no JIT) on GeForce RTX 3090.



一起点亮
GitHub
欢迎大家 Star

jobs.jina.ai



WE
ARE
HIRING

