

PyTorch Seminar

For the beginners

Duhyeon Kim / May 2025



Orientation

2025.04.30



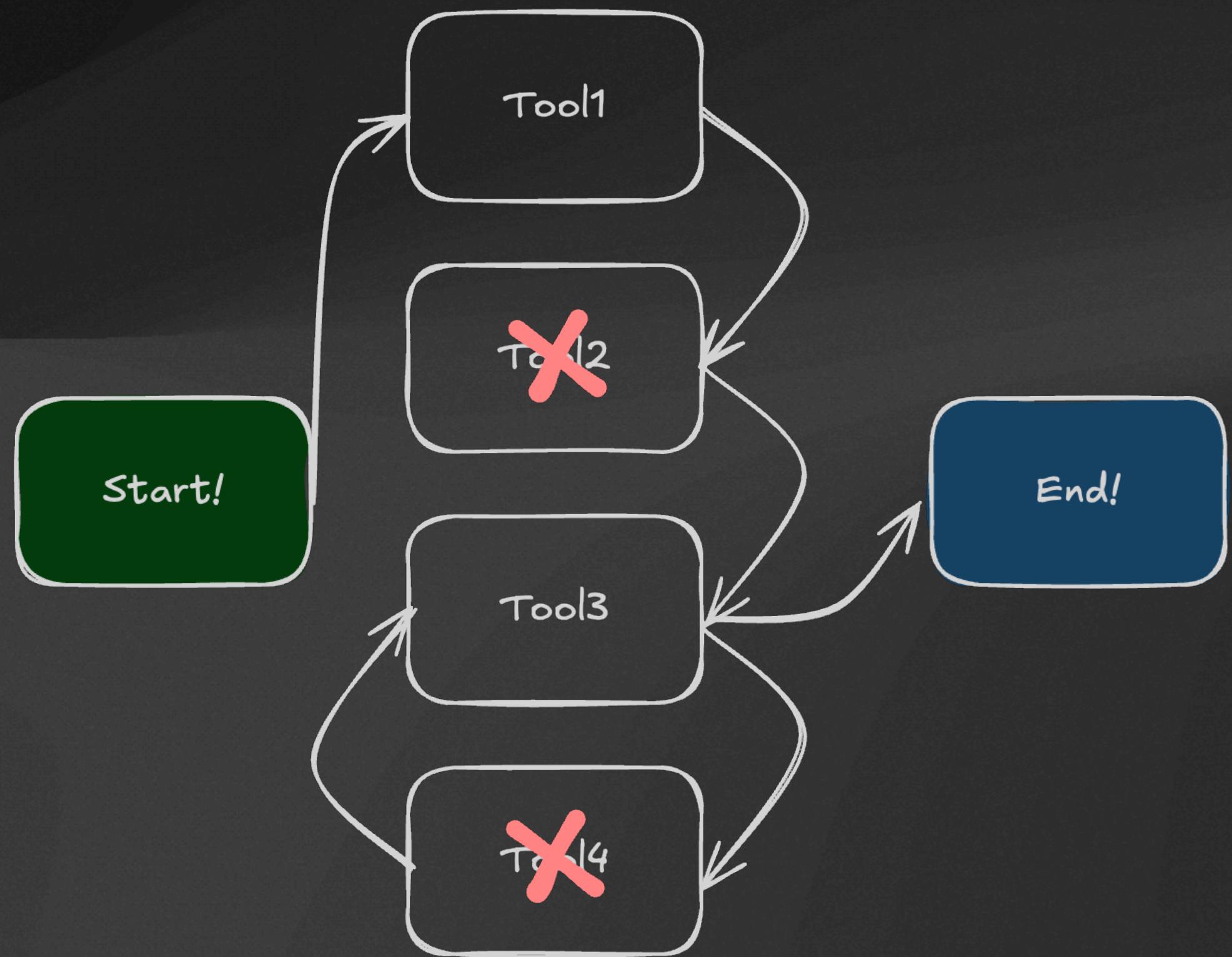
Contents

- A. Why having seminar about PyTorch?
- B. What is Machine Learning Framework?
- C. What are some ML models?
- D. The common Parts of all the Models
- E. What will we focus and What will we be learned?

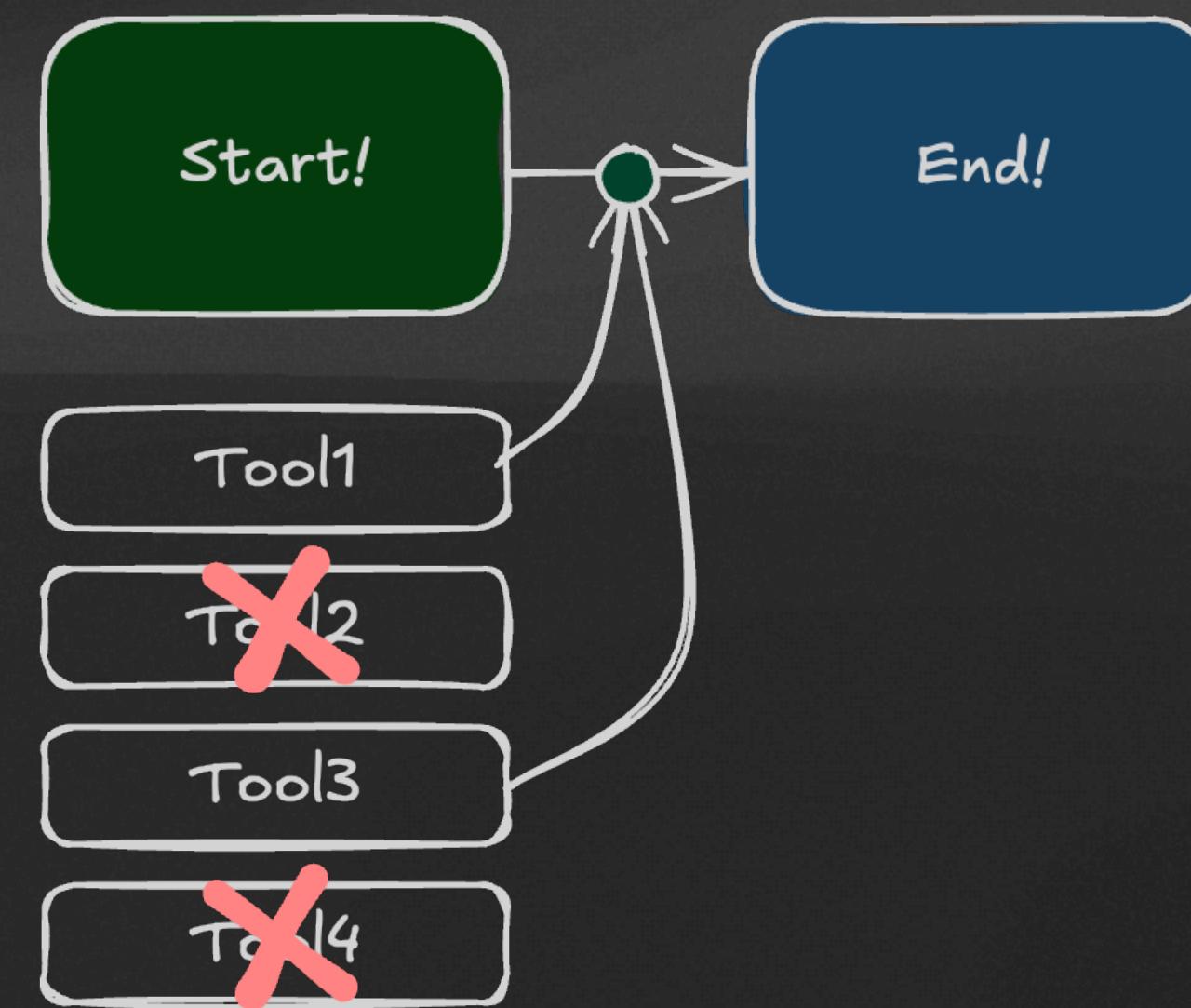
Why having seminar about PyTorch?

Giving the overview and tools to use in later

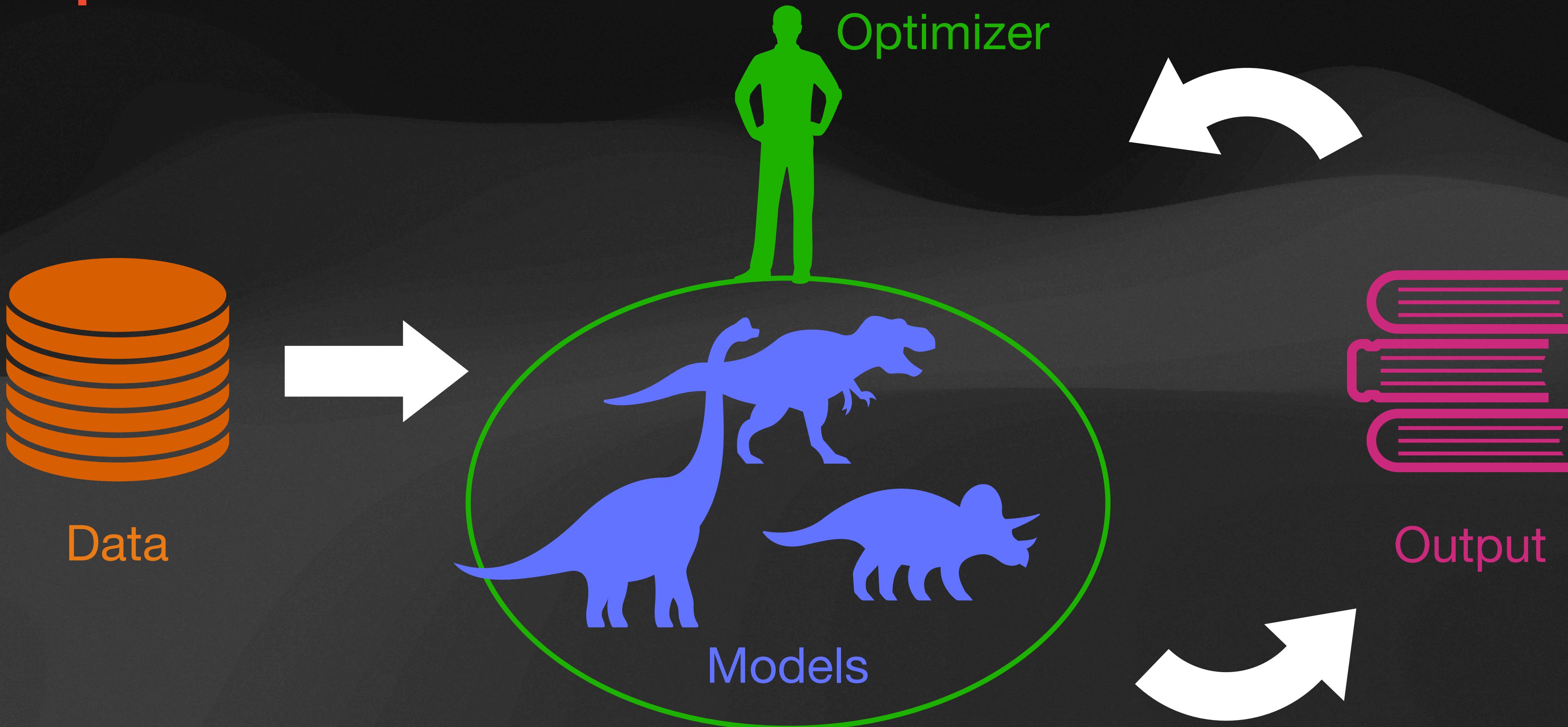
Former Learning Graph



After the Seminar



Abstract Level View of ML Concepts



What is Machine Learning Framework?

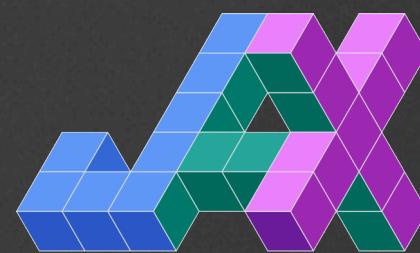
Tensorflow, PyTorch, JAX



```
Conv2D(32, (3, 3), strides=(2, 2), padding='same', use_bias=True)
```



```
nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, stride=1, padding=0)
```



```
nn.Conv(features=32, kernel_size=(3, 3), strides=(1, 1), use_bias=True)
```

```
static Tensor compute_columns2d(
    const Tensor& input,
    IntArrayRef padding,
    IntArrayRef stride,
    IntArrayRef kernel_size,
    bool is_channels_last) {
    const int64_t kernel_height = kernel_size[0];
    const int64_t kernel_width = kernel_size[1];
    const int64_t pad_height = padding[0];
    const int64_t pad_width = padding[1];
    const int64_t stride_height = stride[0];
    const int64_t stride_width = stride[1];
    const int64_t batch_size = input.size(0);
    const int64_t n_input_plane = input.size(1);
    const int64_t input_height = input.size(2);
    const int64_t input_width = input.size(3);
    const int64_t output_height = (input_height + 2 * pad_height - kernel_height) / stride_height + 1;
    const int64_t output_width = (input_width + 2 * pad_width - kernel_width) / stride_width + 1;

    Tensor columns;
    if ((kernel_height == 1) && (stride_height == 1) && (pad_height == 0) &&
        (kernel_width == 1) && (stride_width == 1) && (pad_width == 0)) {
        // Columns are just a view on the input for the 1x1 kernel special case.
        if (is_channels_last) {
            columns = input.as_strided({batch_size, output_height * output_width, n_input_plane},
                                       {output_height * output_width * n_input_plane, n_input_plane, 1}).detach();
        } else {
            columns = input.view({batch_size, n_input_plane, output_height * output_width}).detach();
        }
    ...
}
```

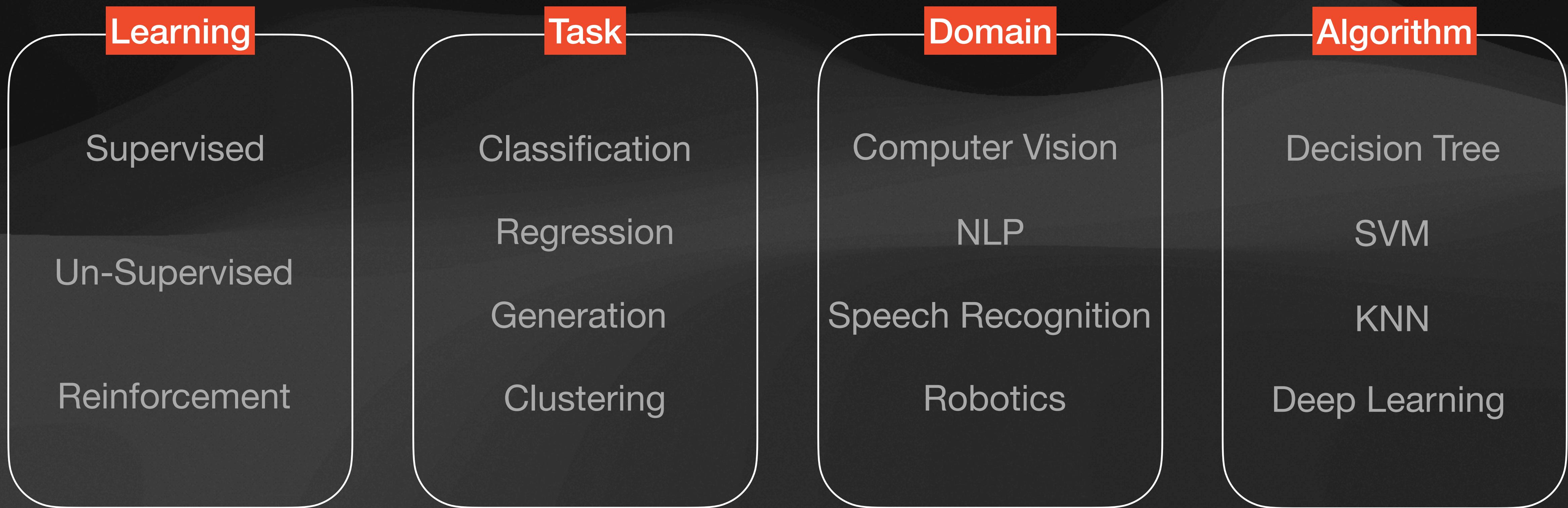


PyTorch

Source code : <https://github.com/pytorch/pytorch/blob/main/aten/src/ATen/native/ConvolutionMM2d.cpp>

How Machine Learning is categorized?

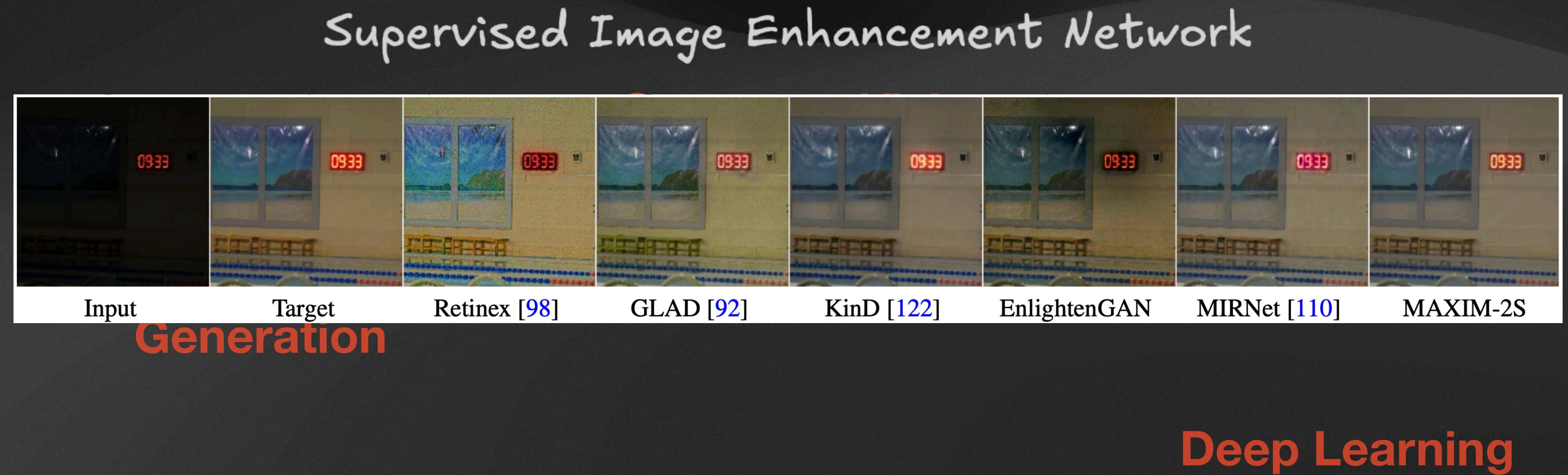
Learning, Task, App Domain, Algorithm



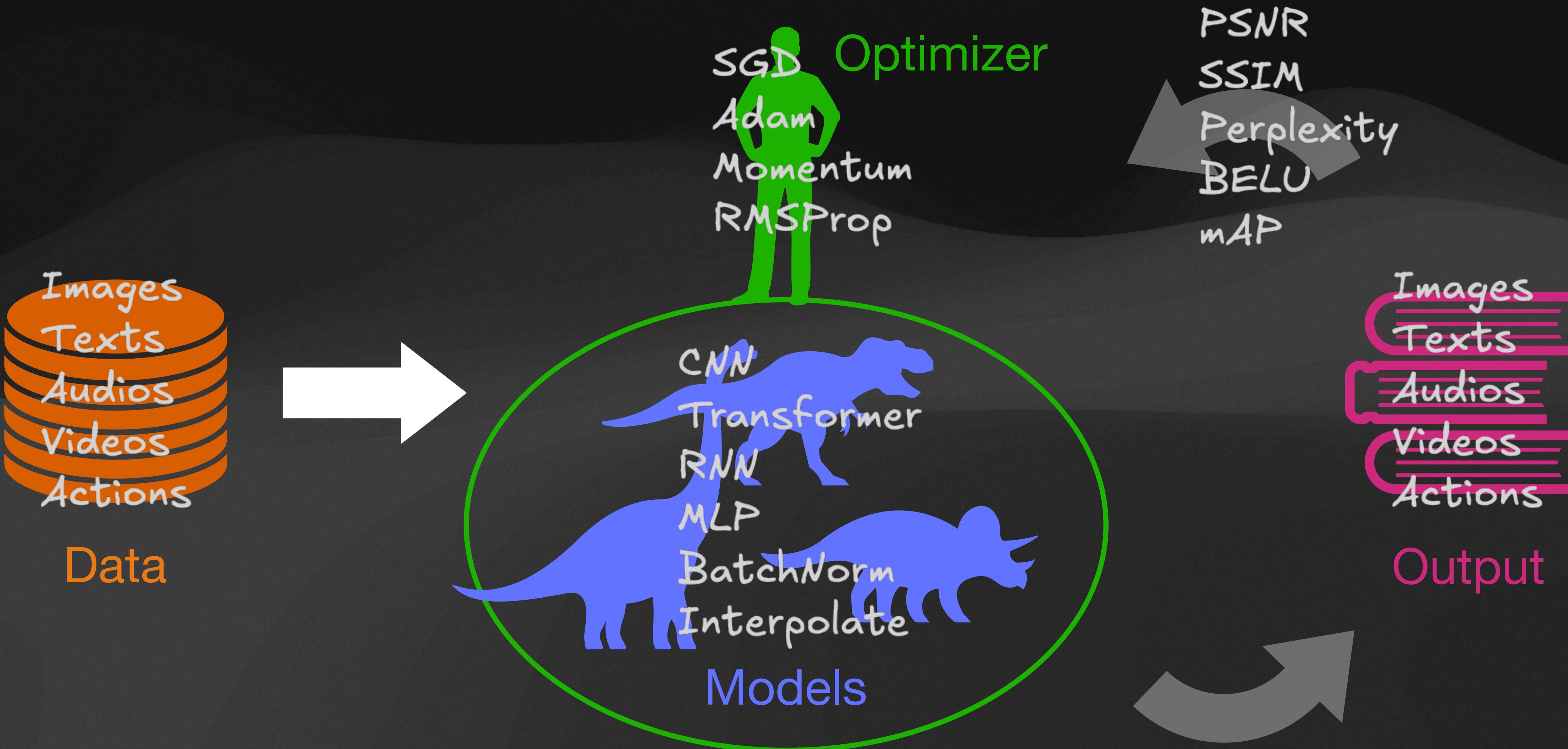
How Machine Learning is categorized?

Learning, Task, App Domain, Algorithm

Supervised
Generation
Computer Vision
Deep Learning



The common Parts of all the Models



What will we focus and What will we be learned?

- Basic Training Code Structure
- Analyzing each Training Block (e.g. Dataloader.py, Util.py, Model.py)
- Simple Supervised Image Classifier Training
- Other View of Model
- Neural Style Transfer (NST)
- The importance of Loss function

The background features a minimalist design with abstract, undulating shapes. The lower half is dominated by large, soft-edged waves in a bright orange-yellow color. Above these, there are several layers of smaller, darker purple and maroon waves that recede into the distance, creating a sense of depth. The overall aesthetic is clean and modern, with a focus on color and form.

Let's go