Machine Learning

RAJESH SHARMA
Walt Disney Animation Studios

Thank you to ACM SIGGRAPH!

Pol Jeremias-Vila : SIGGRAPH 2021 Chair

Tomasz Bednarz: Frontiers Program Chair

Alex Bryant: Student Volunteers Chair

Tim Hendrickson: Digital Marketing Manager

Student Volunteers:
Rogelio, Trinity, Aurora, Emily, Hunter & Kendra

SIGGRAPH 2021

3

Machine Learning

Rajesh Sharma

Marios Papas

Research Scientist

Research Scientist

Disney Research | Studios - Zurich

As the Rendering Group lead at Disney Research, Marios works with a talented team of researchers, engineers and students from Walt Disney Animation Studios, Pixar, Industrial Light and Magic and ETH Zurich.

His research focuses on Monte Carlo rendering problems to understand how light interacts with materials and use that knowledge to design state-of-the-art algorithms and appearance models for efficient and realistic image synthesis.

His current research focuses on using Machine Learning methods for information sharing between samples generated by path tracing algorithms for driving future sampling decisions (adaptive importance sampling) and achieve low-error reconstruction (denoising).

Today

- Quick Recap: Regression, HW
- Second Neural Network - Classification
- Least Squares → Maximum Likelihood
- Autoencoder

Hands-on

- ★ Log in to your google drive
- ★ Make a shortcut to: `https://bit.ly/3oKCVCh`
- ★ Make a copy of:
  - ○ Sinx.ipynb (Homework from last time)
  - ○ FlowerClassification.ipynb
  - ○ Autoencoder.ipynb

Recap - Questions

Aidan Whelan - What kinds of models do you find are most commonly used in your work in the film industry?
Avidsiman - How do we know if the model has memorized our data instead of fitted it?
Anjoe Jacob - Can you also expand a bit about factors that go into choosing between mean avg err vs mean squared err?

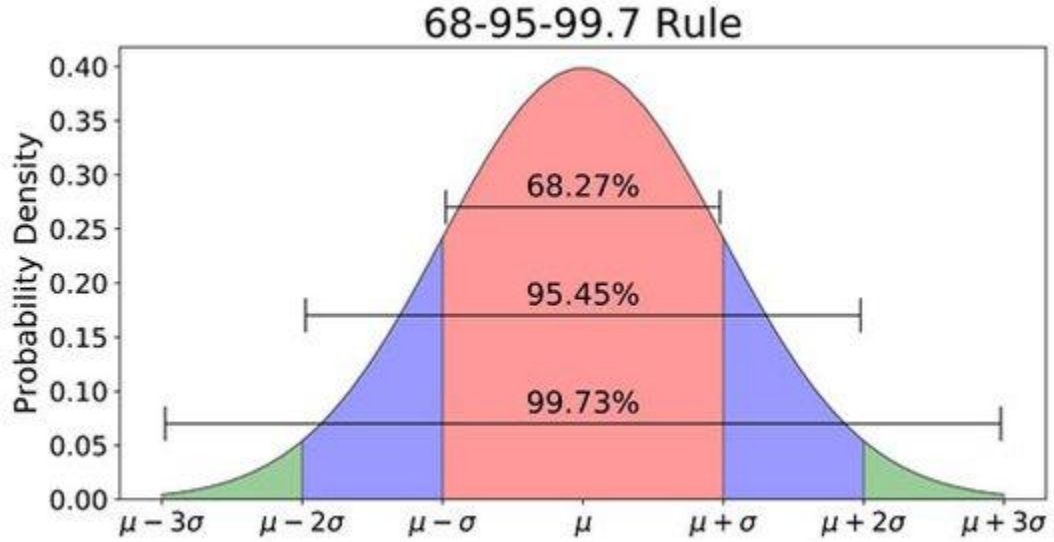
Classification

Given some pre-classified data:

Q: How do we predict which class the new data belongs to?

A: By finding the maximum likelihood

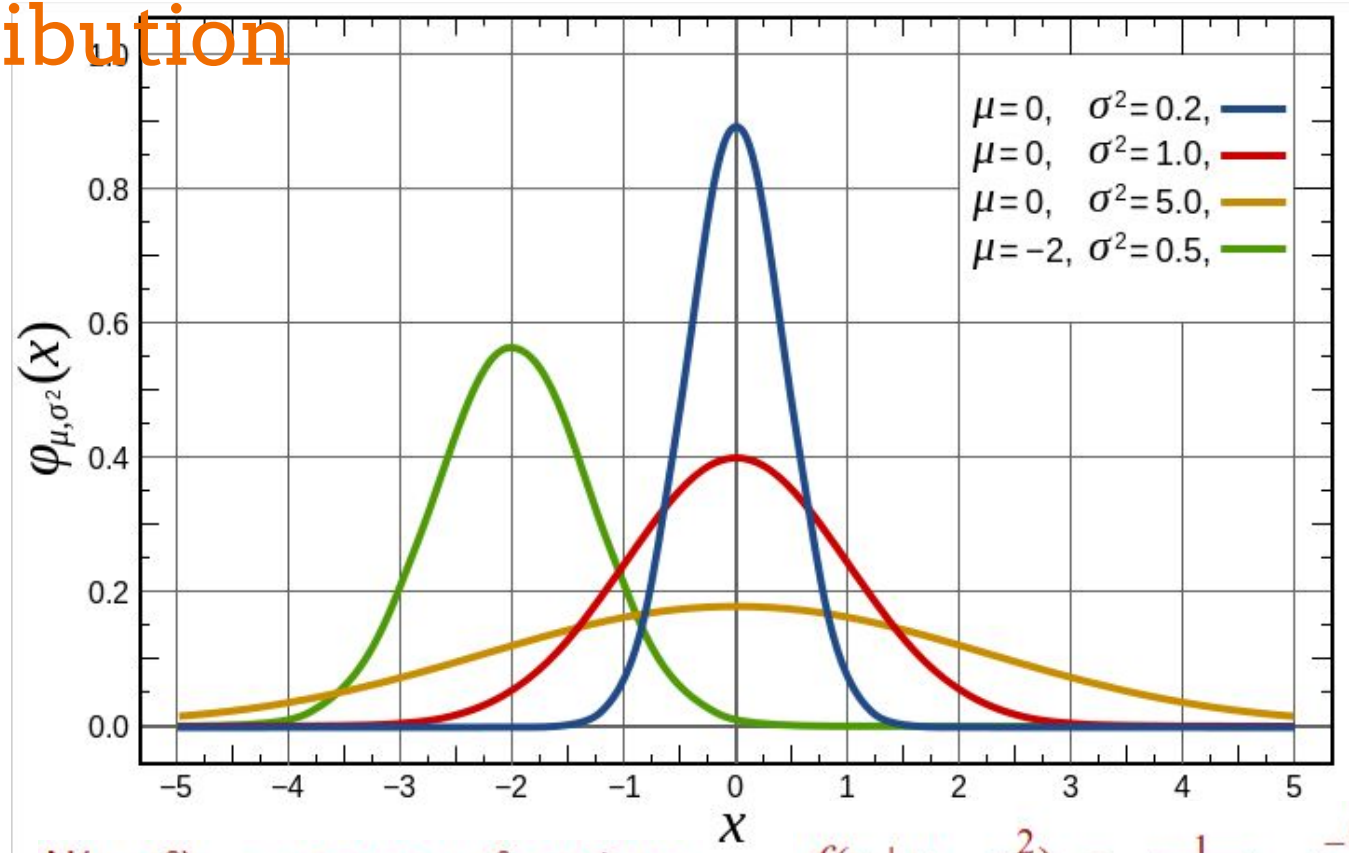
Probability & Statistics: Normal Distribution



$N(\mu, \sigma^2)$; μ - mean , σ^2 - variance

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

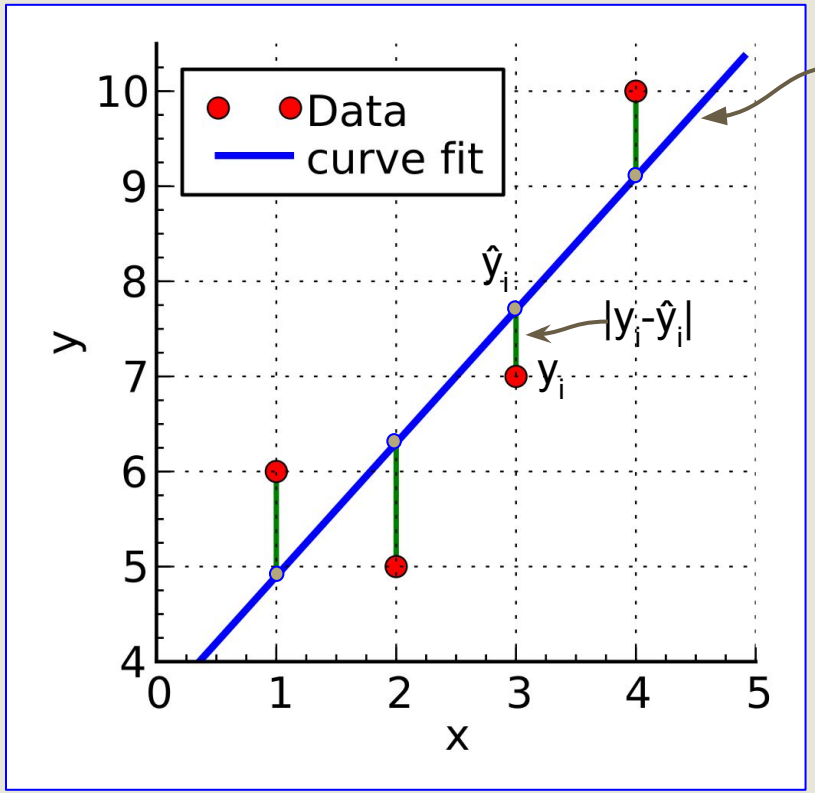
Probability & Statistics: Normal Distribution



$N(\mu, \sigma^2)$; μ - mean , σ^2 - variance

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Regression → Maximum Likelihood Estimation



Prediction: $\hat{y} = ax + b$

Actual: y_i

Error: $|y_i-\hat{y}_i|$

Total Squared Error:

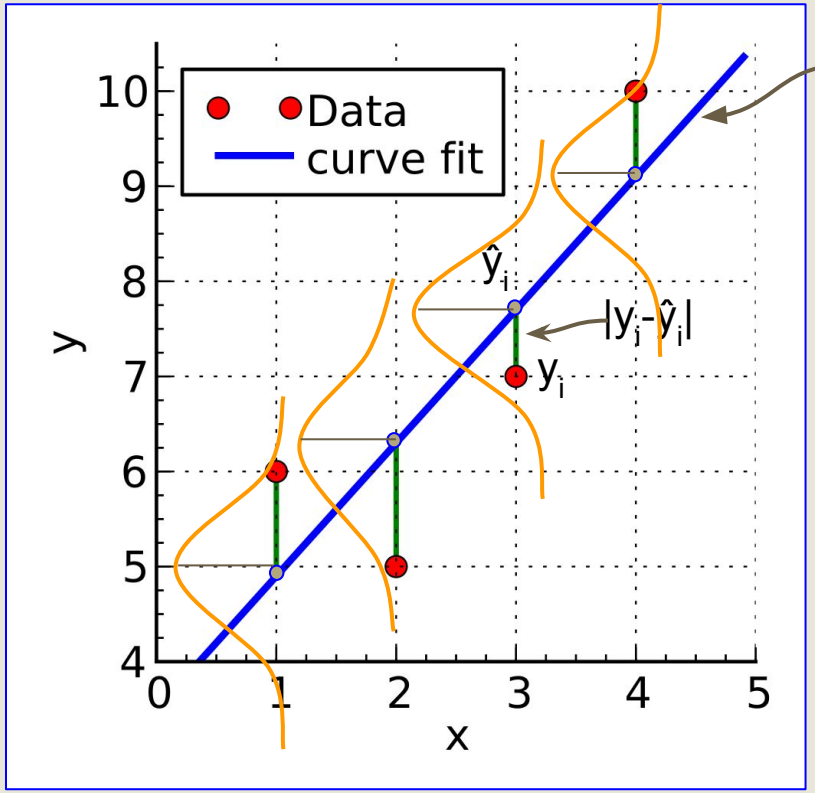
$\sum (y_i-\hat{y}_i)^2$, for i=(1, n)

Minimize Total Squared Error:

$E(a,b) = \sum(y_i - ax_i - b)^2$

(a,*b*) are the parameters (weights)

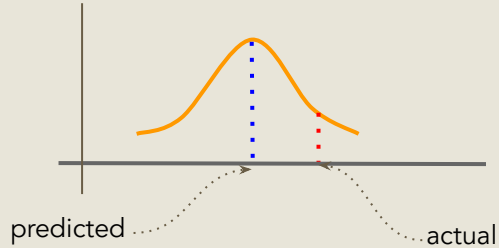
Regression → Maximum Likelihood Estimation



Prediction: $\hat{y} = ax + b$

Actual: y_i

Error: $|y_i-\hat{y}_i|$



Maximum Likelihood Estimation vs MSE

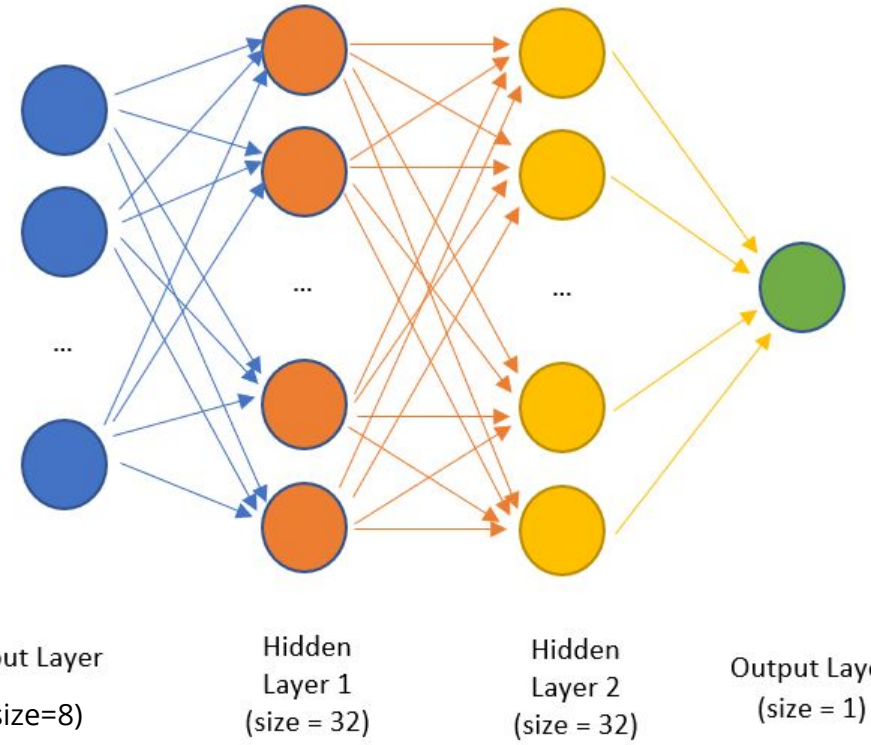
$$f(x \mid \mu, \ \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$f(x_1, \ldots, x_n \mid \mu, \sigma^2) = \prod_{i=1}^{n} f(x_i \mid \mu, \sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(-\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{2\sigma^2}\right).$$

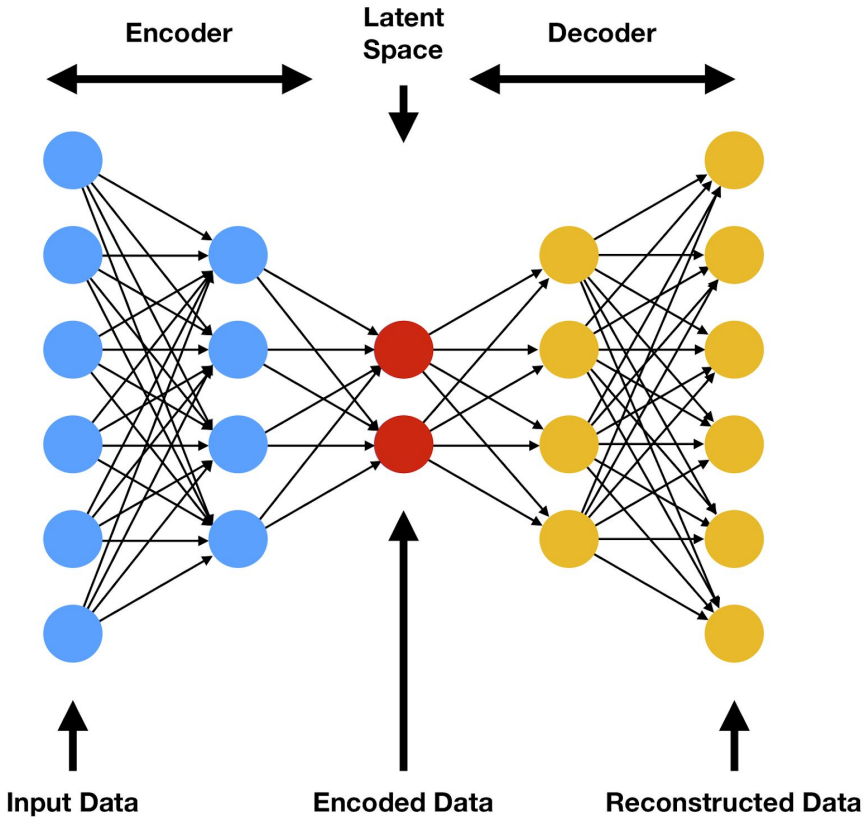
$$\log\Big(\mathcal{L}(\mu, \sigma)\Big) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

Autoencoder

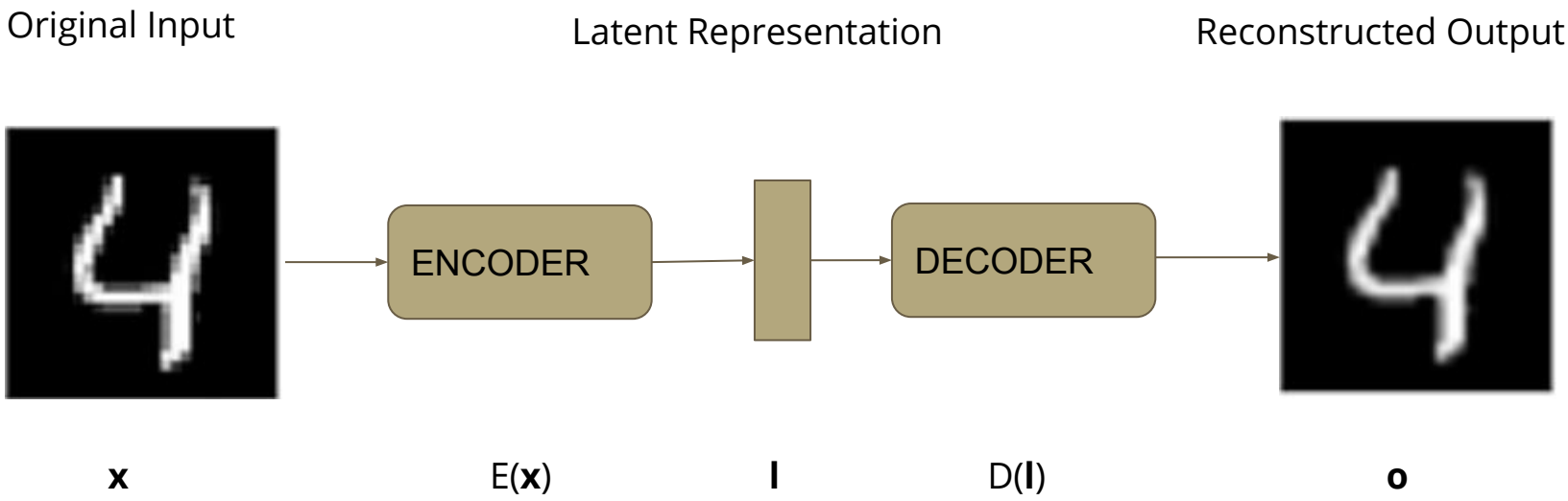
For regression, we had a fully-connected network, output layer size=1



Autoencoder



Autoencoder



Hands-on

- ★ *Log in to your google drive*
- ★ Find the shared folder 'Disney Machine Learning Webinar'
- ★ Make a copy of:
  - AutoEncoder.ipynb

Autoencoder - Model

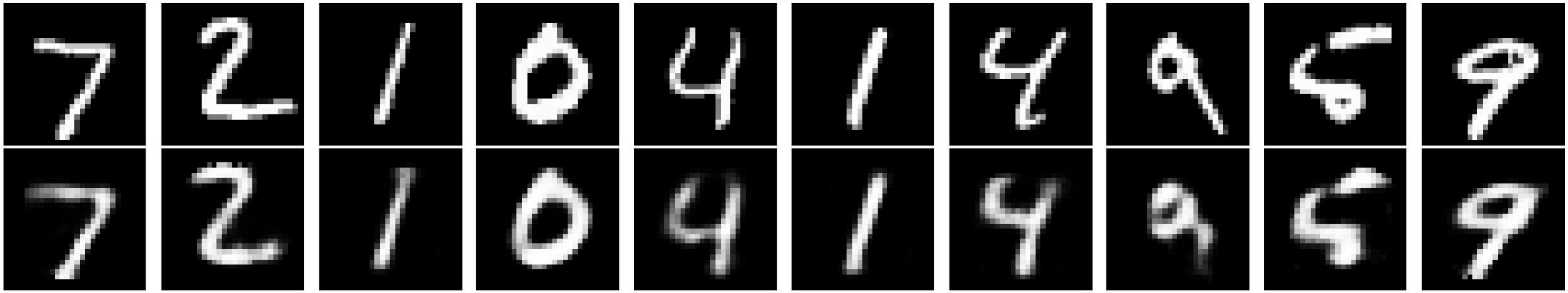
```
# build an autoencoder
model = tf.keras.models.Sequential([
    tf.keras.layers.InputLayer(IMG_SHAPE),
    tf.keras.layers.Flatten(),
    # encoder
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    # decoder
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(784, activation='sigmoid'),
    tf.keras.layers.Reshape(IMG_SHAPE)
    ])

# compile
model.compile(optimizer='adamax', loss='mse')

# fit
model.fit(x_train, x_train, epochs=17,batch_size=256, shuffle=True, validation_data=(x_test, x_test))

# predict
decoded_imgs = model.predict(x_test)
```

Autoencoder - results



Compression Factor: 28x28/32 ~ 25X

Caution

Remarkably Clever

Surprisingly Dumb

Next Class

- Efficient Data Pipeline
- Convolutional Neural Network
- Artistic Style Transfer
- Homework:
  - Use the mnist dataset for classification
  - Extra credit: also show “next likely”
- @xarmalarma, #siggraph2021