Generative AI with Diffusion Models

Part 5: CLIP

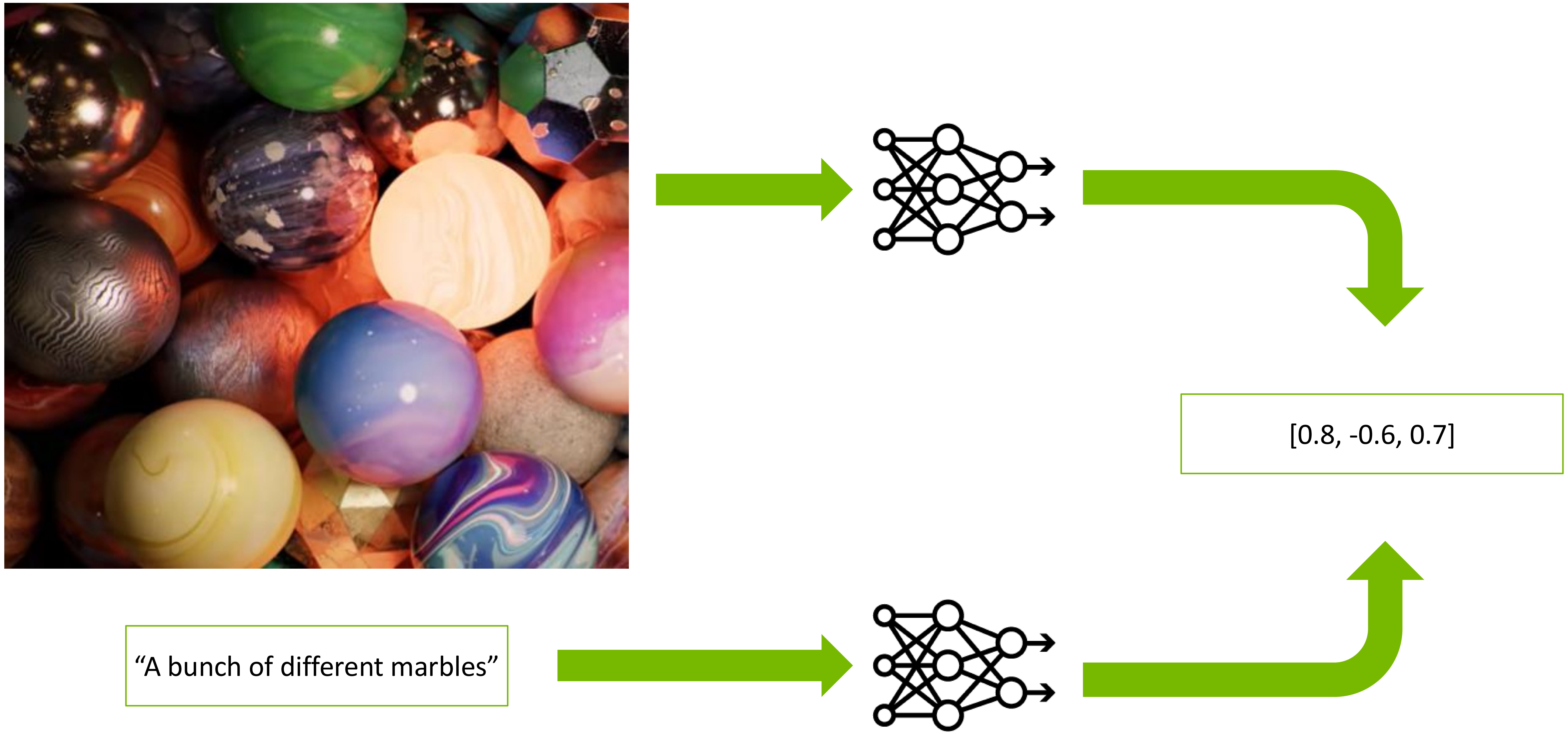
Agenda

- Part 1: From U-Nets to Diffusion
- Part 2: Denoising Diffusion Probabilistic Models
- Part 3: Optimizations
- Part 4: Classifier-Free Diffusion Guidance
- Part 5: CLIP
- Part 6: Wrap-up & Assessment

Contrastive Language-Image Pre-Training (CLIP)

Matching Text to Image

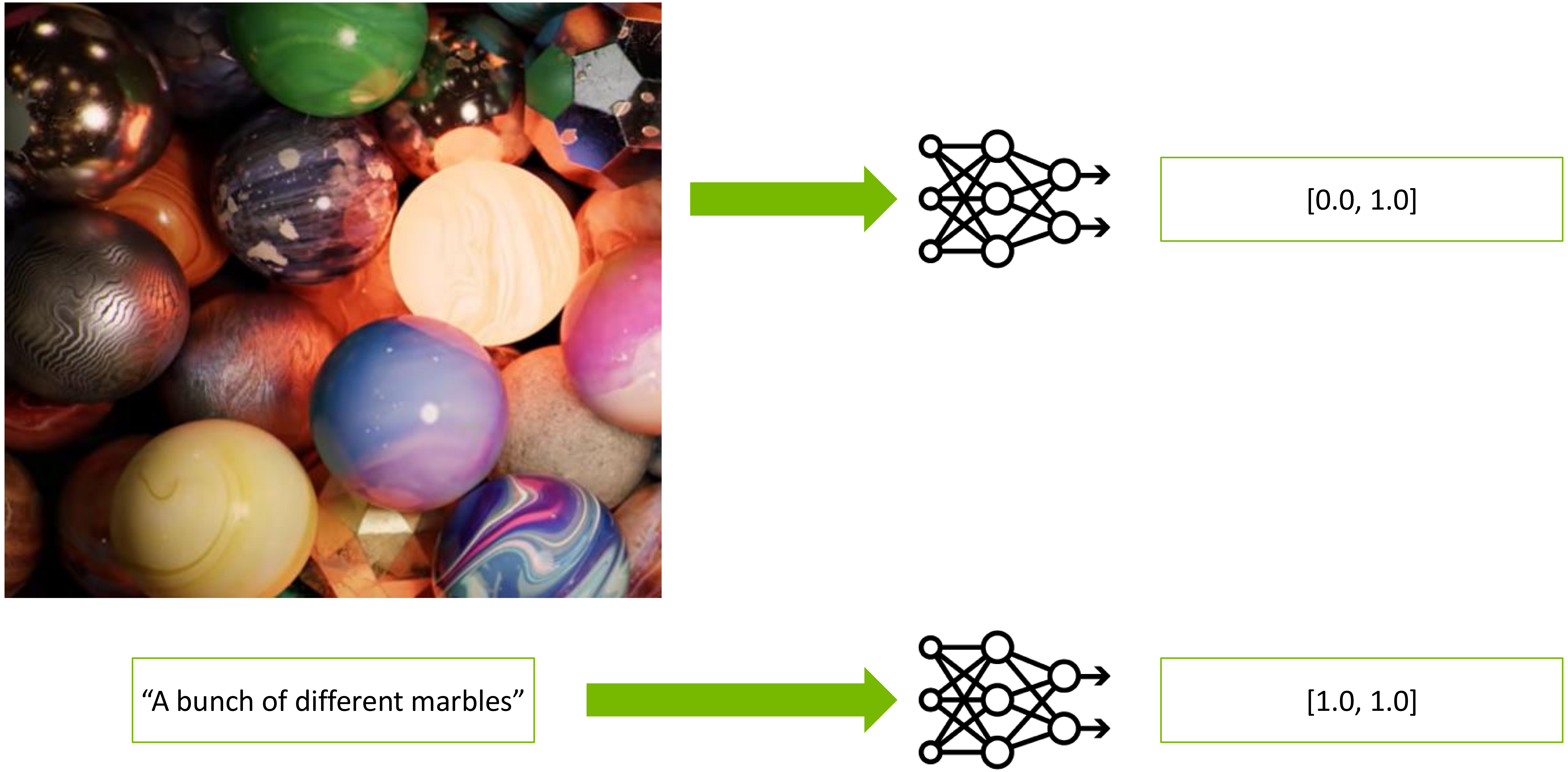
Is it Possible?



“A bunch of different marbles”

[0.8, -0.6, 0.7]

Cosine Similarity

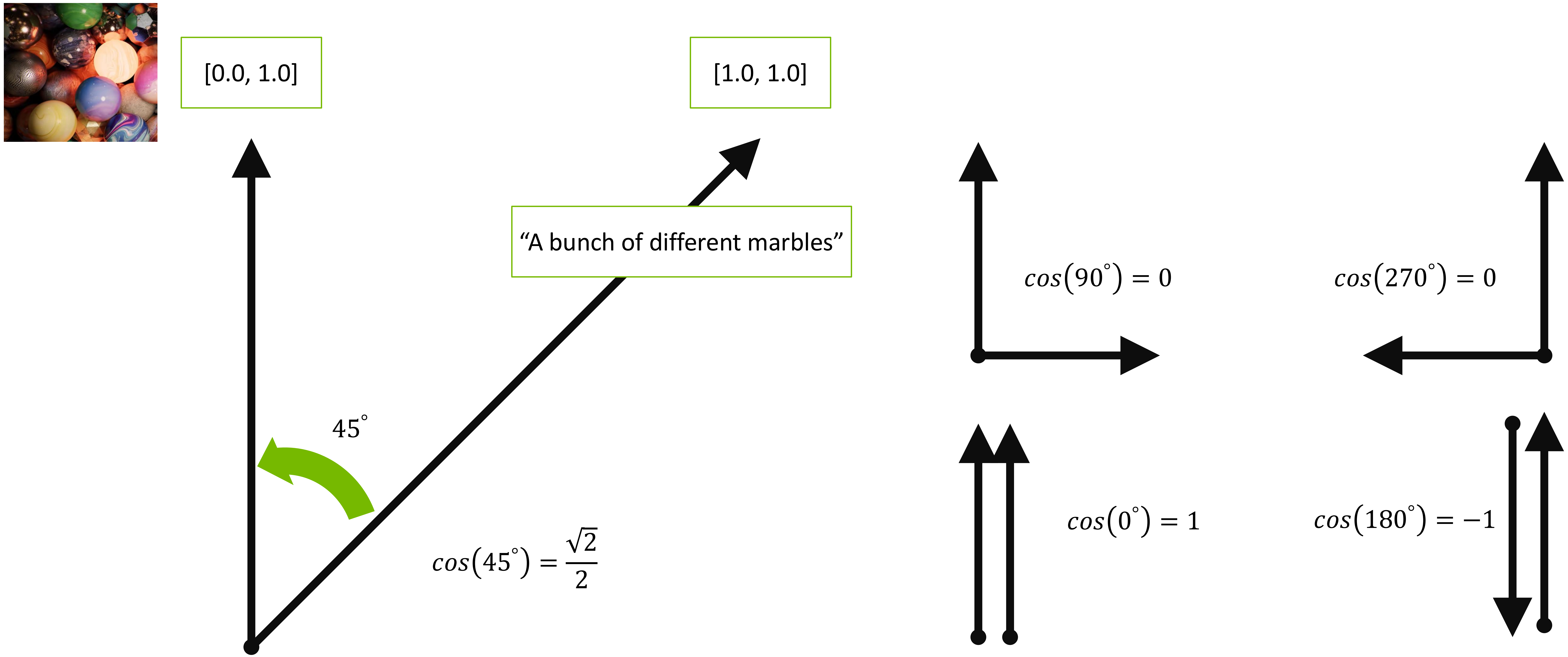


[0.0, 1.0]

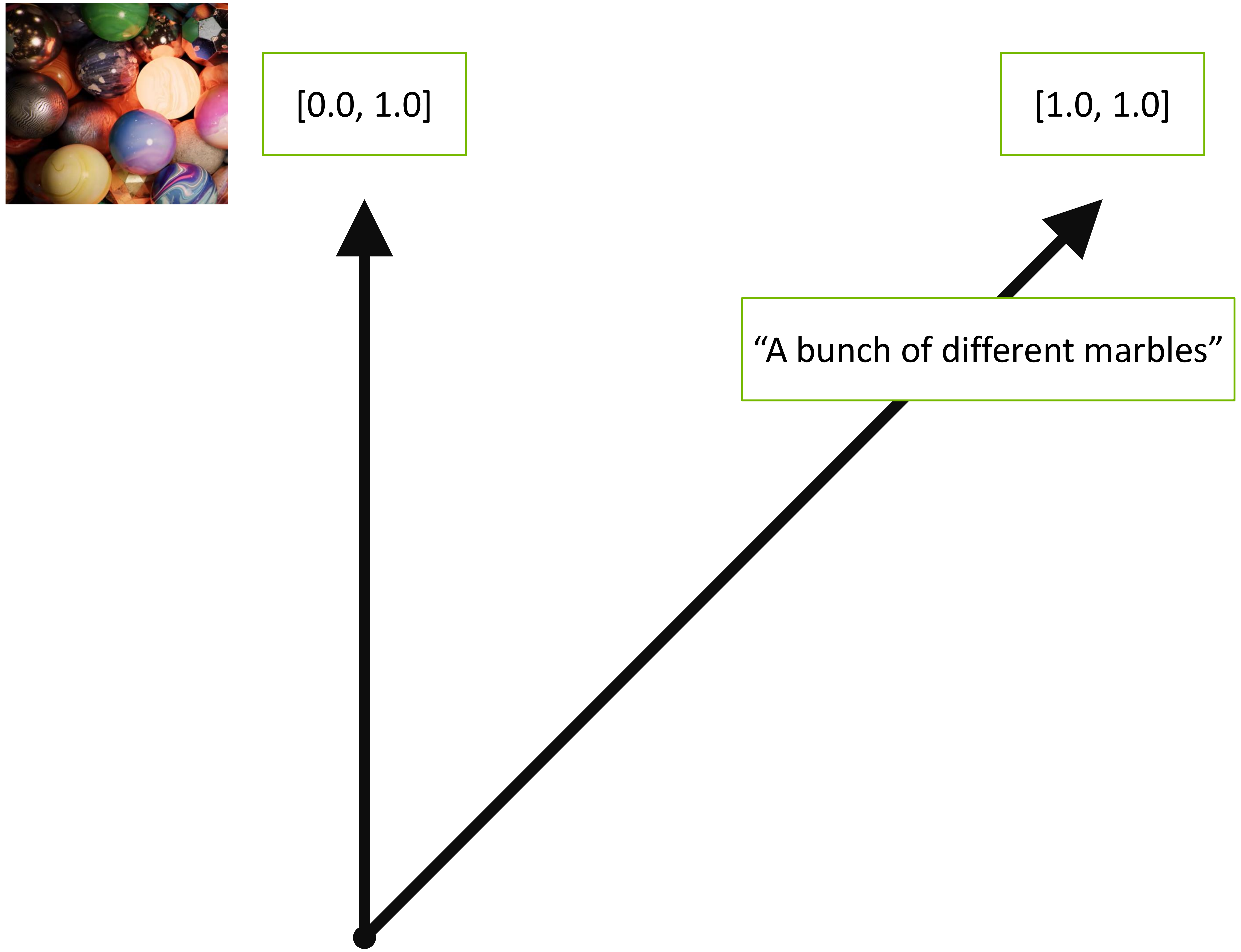
“A bunch of different marbles”

[1.0, 1.0]

Cosine Similarity



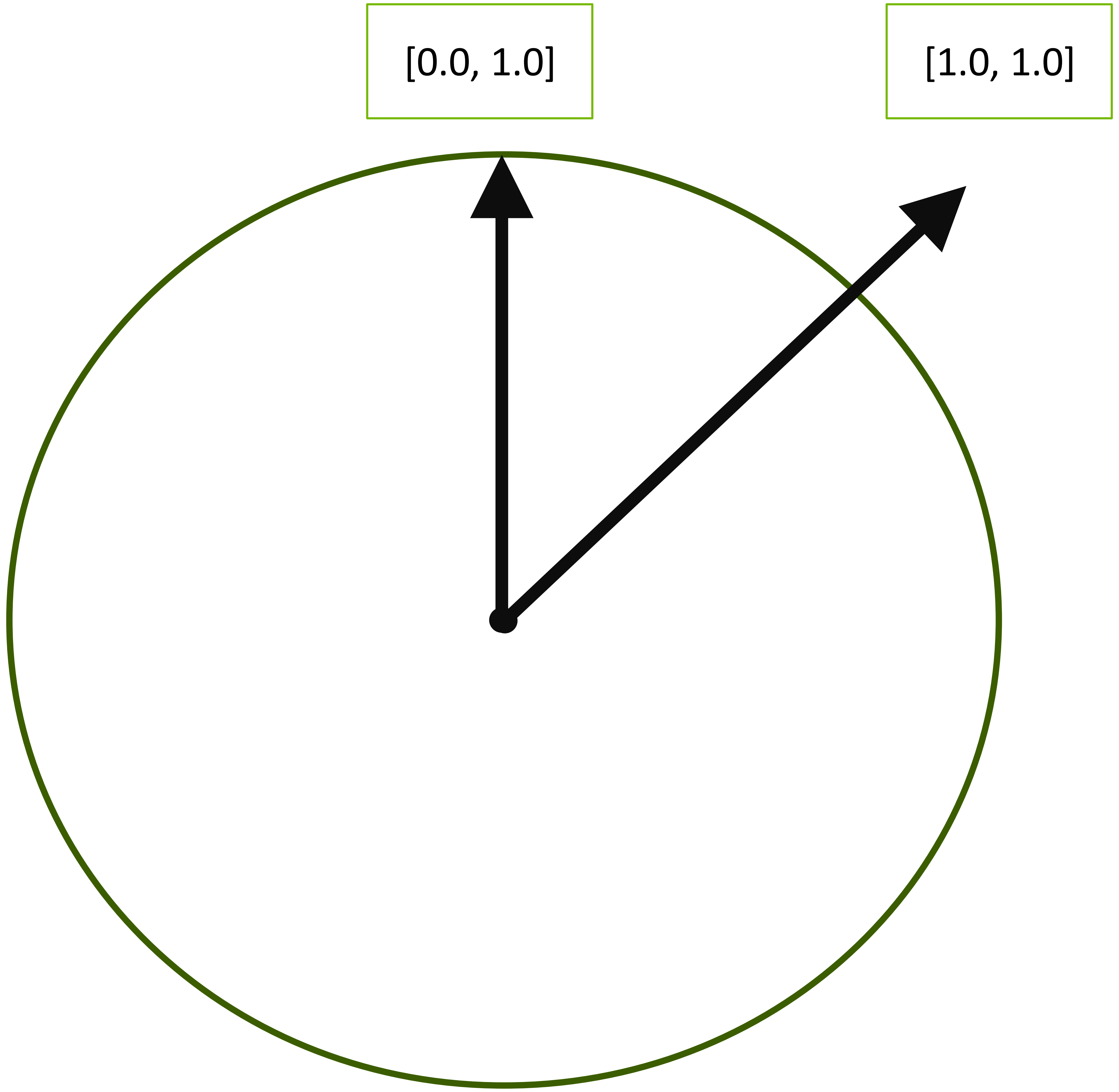
Dot Product



[0.0, 1.0]

[1.0, 1.0]

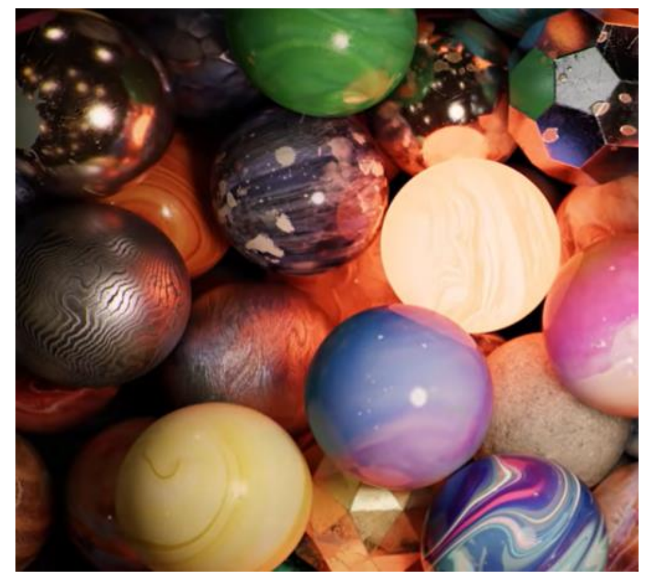
“A bunch of different marbles”

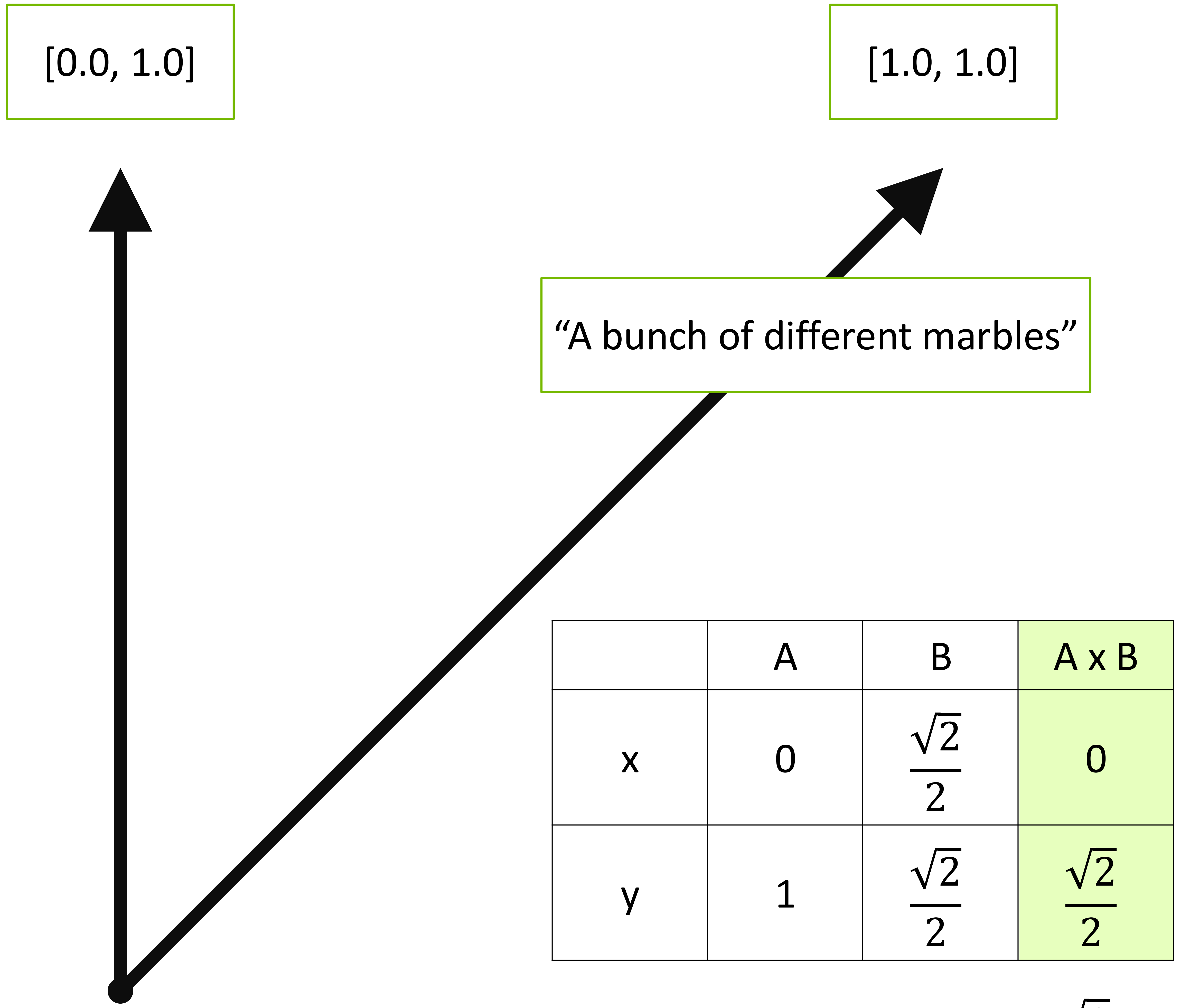


[0.0, 1.0]

[1.0, 1.0]

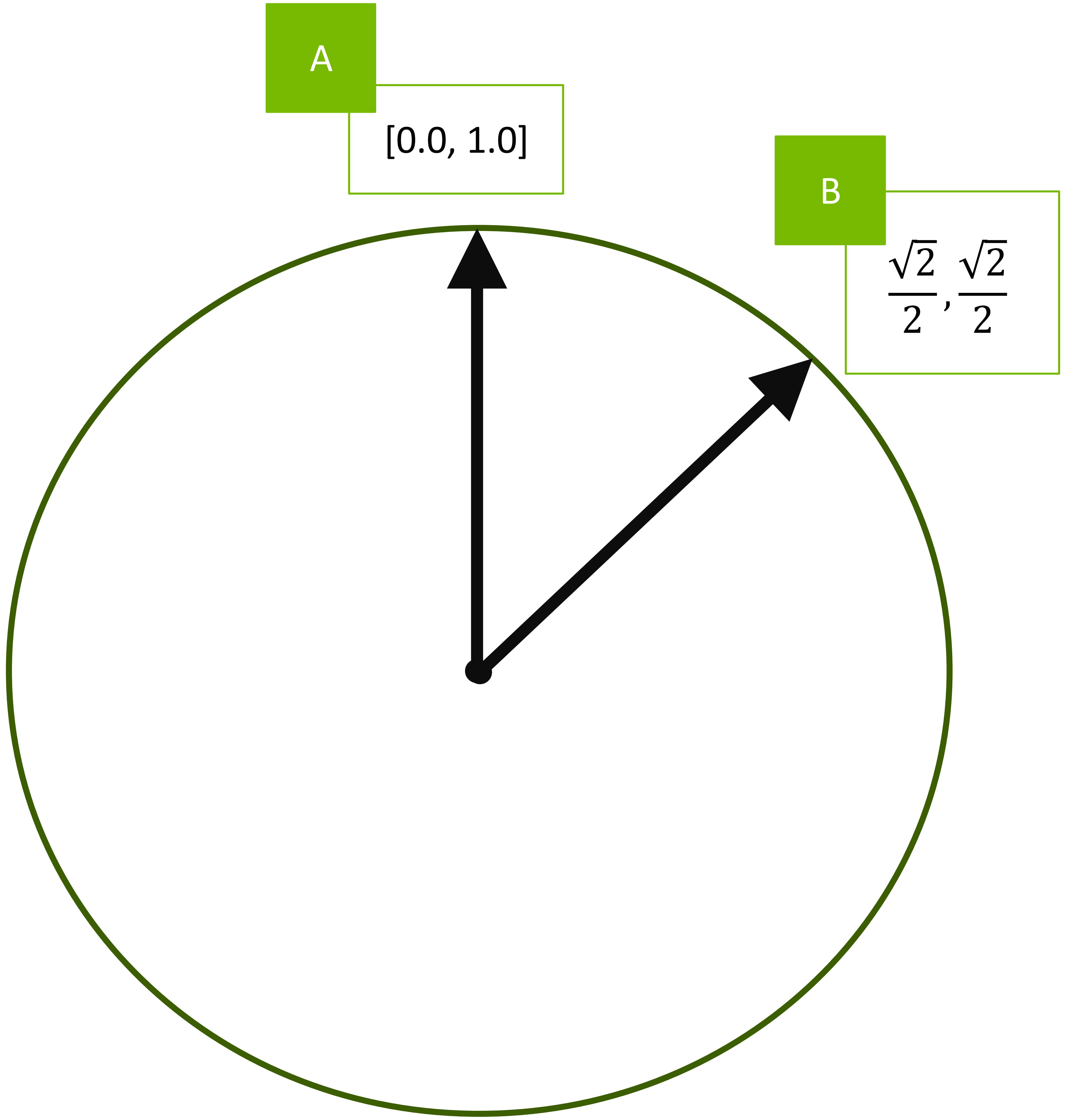
Dot Product



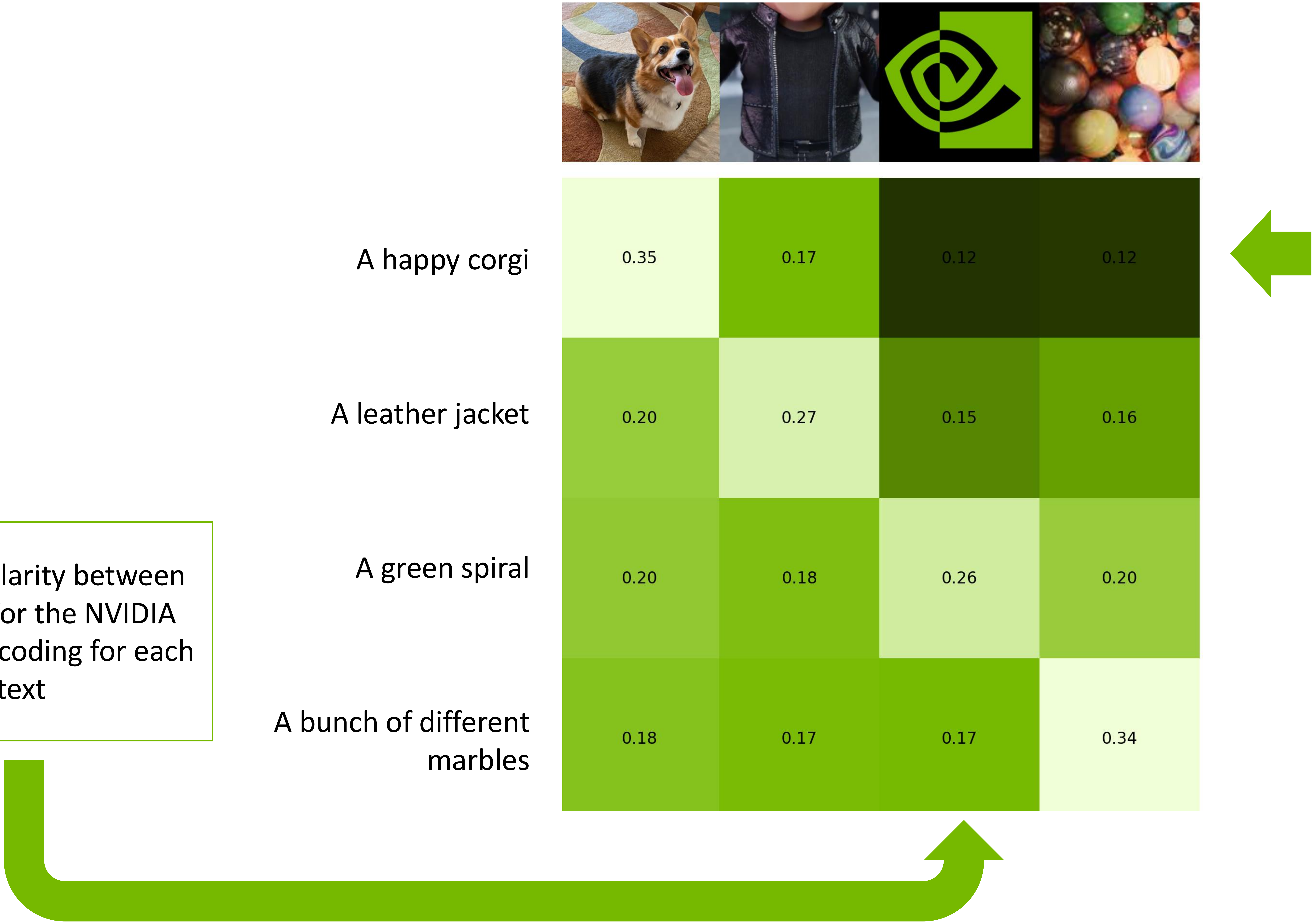


	A	B	A x B
x	0	$\frac{\sqrt{2}}{2}$	0
y	1	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$

$$cos(45^{\circ}) = \frac{\sqrt{2}}{2}$$



CLIP Training

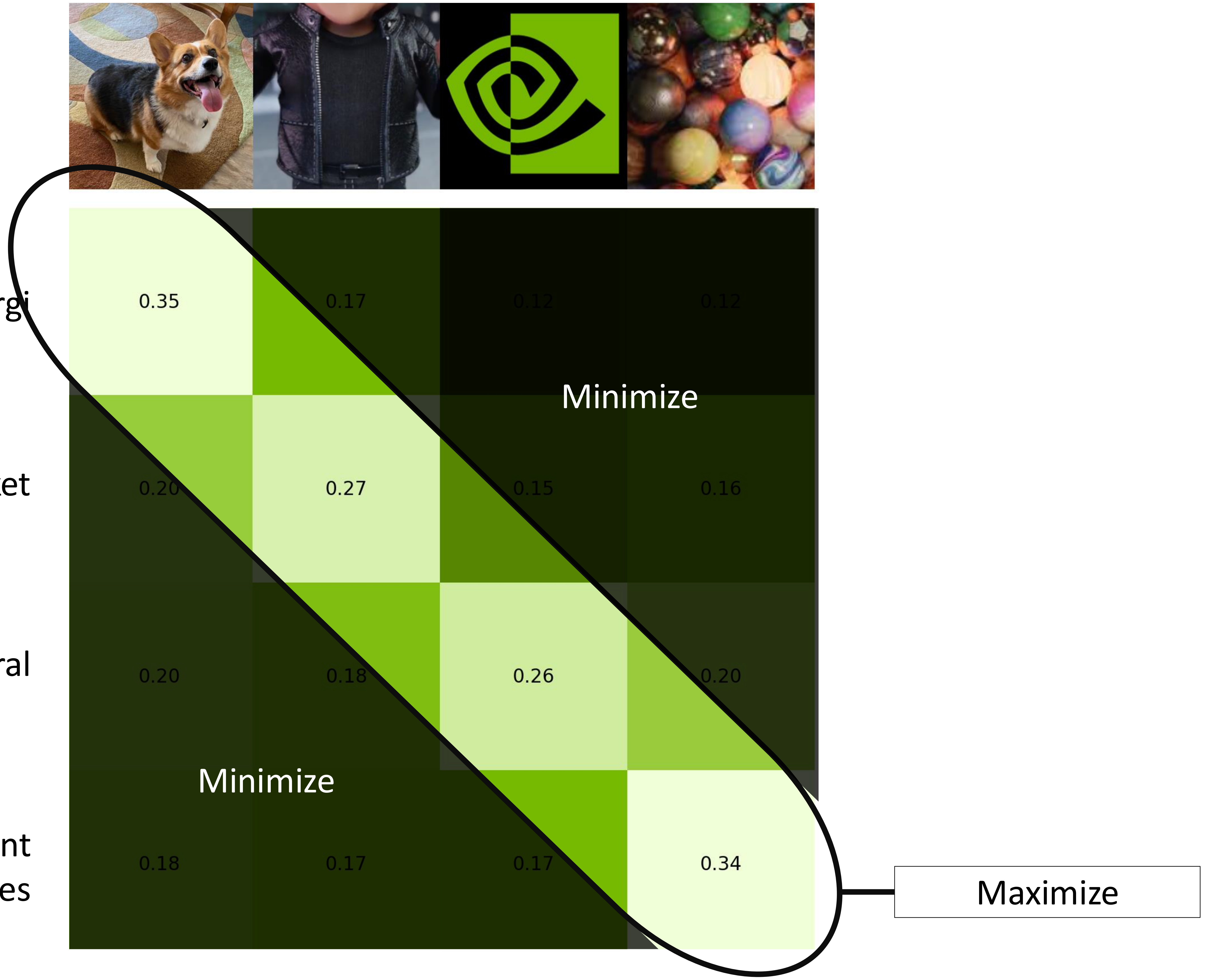


A happy corgi	0.35	0.17	0.12	0.12
A leather jacket	0.20	0.27	0.15	0.16
A green spiral	0.20	0.18	0.26	0.20
A bunch of different marbles	0.18	0.17	0.17	0.34

Cosine similarity between encoding for "A happy corgi" and encoding for each image

Cosine similarity between encoding for the NVIDIA logo and encoding for each text

CLIP Training



A happy corgi	0.35	0.17	0.12	0.12
A leather jacket	0.20	0.27	0.15	0.16
A green spiral	0.20	0.18	0.26	0.20
A bunch of different marbles	0.18	0.17	0.17	0.34

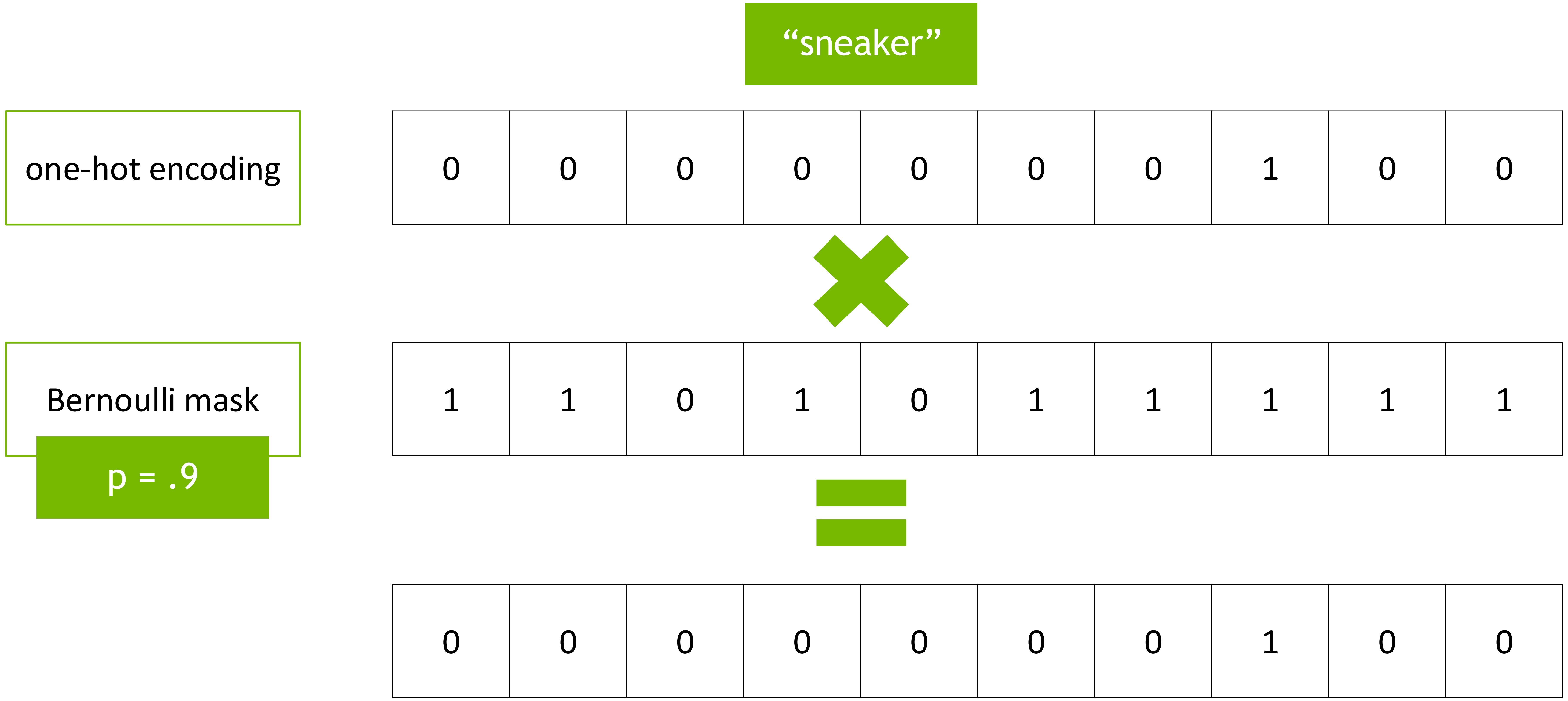
An Experiment

> If CLIP is a pretrained model, do we need text labels to make a text-to-image model?

The Final Model

Same as before
Time
Position Embed
Time Embed 2
Feature Map (Down0)
Down0 Copy
Feature Map (Up2)
Time Embed 1
Context Embed 2
Feature Map (Down1)
Down1 Copy
Feature Map (Up1)
Context Embed 1
Feature Map (Down2)
Down2 Copy
Feature Map (Up0)
Context
Latent Vector

From Class to Context



“sneaker”

one-hot encoding

0	0	0	0	0	0	0	1	0	0

×

Bernoulli mask

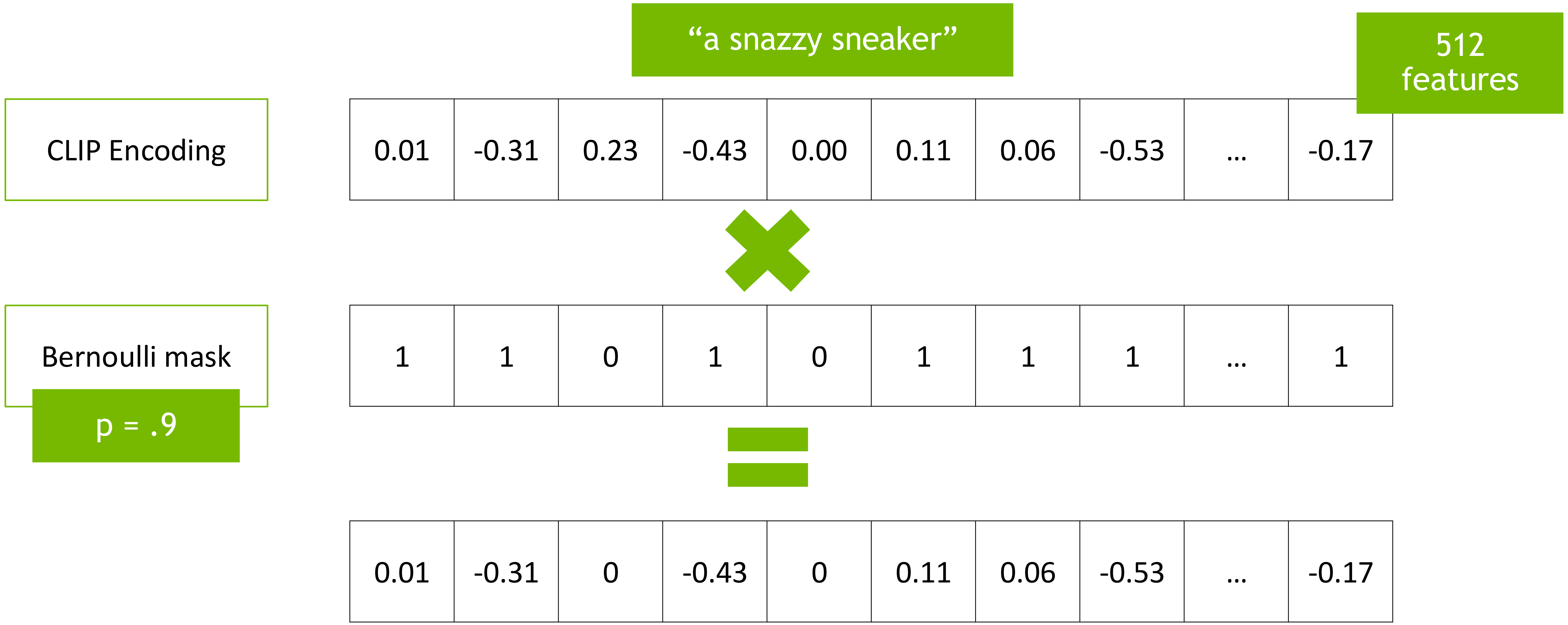
p = .9

1	1	0	1	0	1	1	1	1	1

=

0	0	0	0	0	0	0	1	0	0

From Class to Context

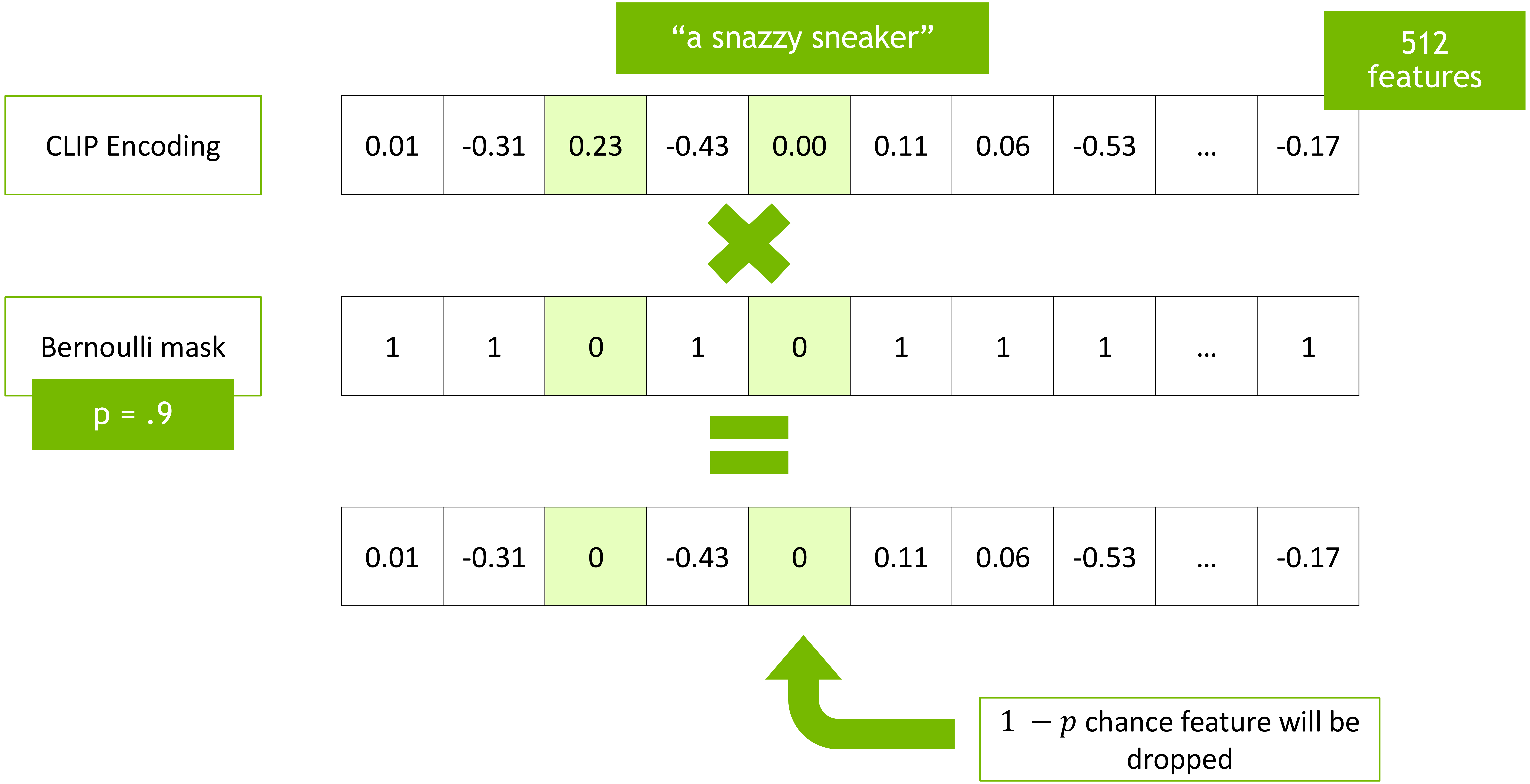


“a snazzy sneaker”

512 features

CLIP Encoding	0.01	-0.31	0.23	-0.43	0.00	0.11	0.06	-0.53	…	-0.17
	×									
Bernoulli mask p = .9	1	1	0	1	0	1	1	1	…	1
	=									
	0.01	-0.31	0	-0.43	0	0.11	0.06	-0.53	…	-0.17

From Class to Context



“a snazzy sneaker”

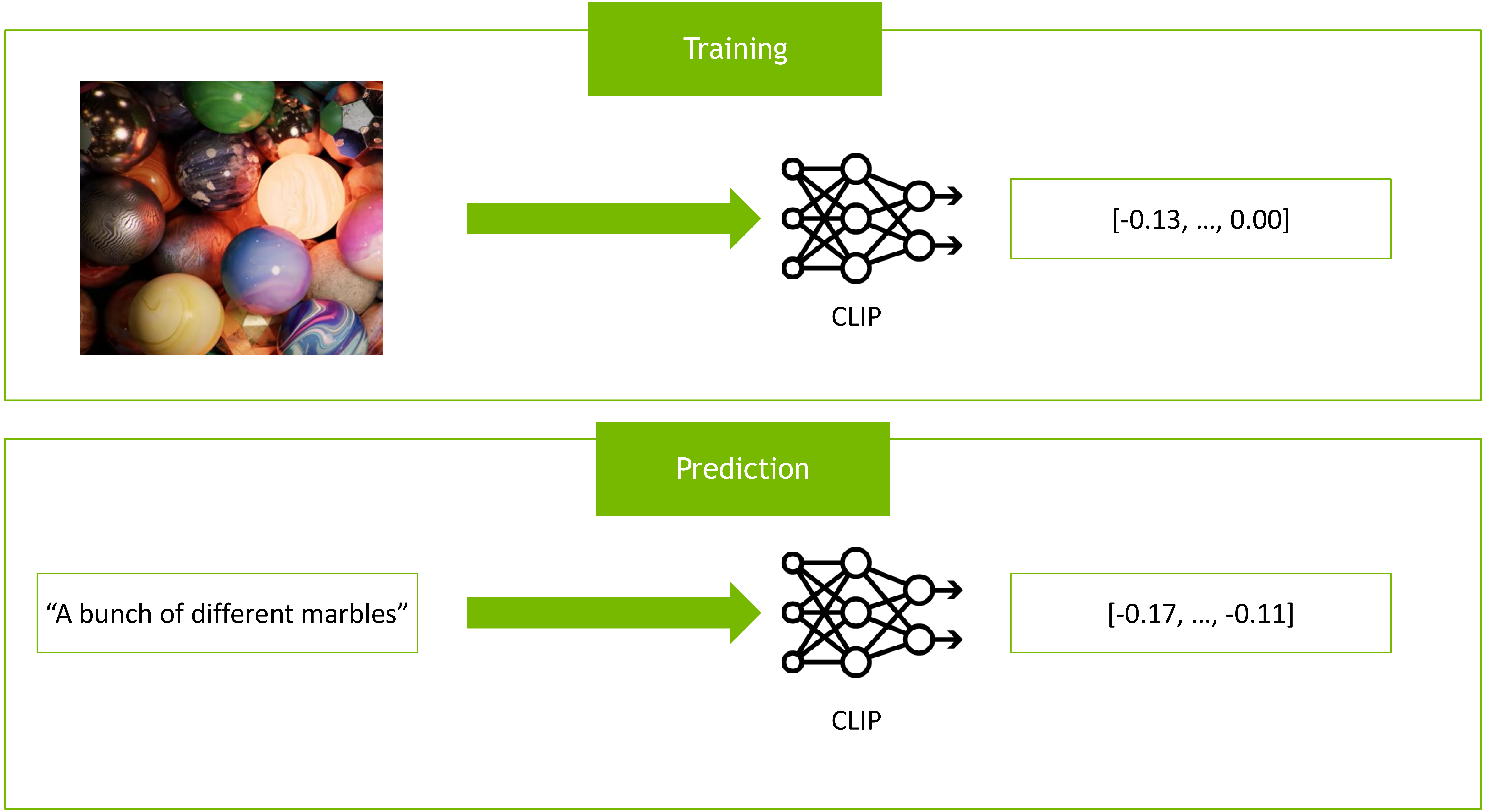
512 features

CLIP Encoding	0.01	-0.31	0.23	-0.43	0.00	0.11	0.06	-0.53	…	-0.17
Bernoulli mask	1	1	0	1	0	1	1	1	…	1
	0.01	-0.31	0	-0.43	0	0.11	0.06	-0.53	…	-0.17

p = .9

$1 - p$ chance feature will be dropped

Experimenting with CLIP



Training

CLIP

[-0.13, …, 0.00]

Prediction

“A bunch of different marbles”

CLIP

[-0.17, …, -0.11]

Let’s get started!

NVIDIA®