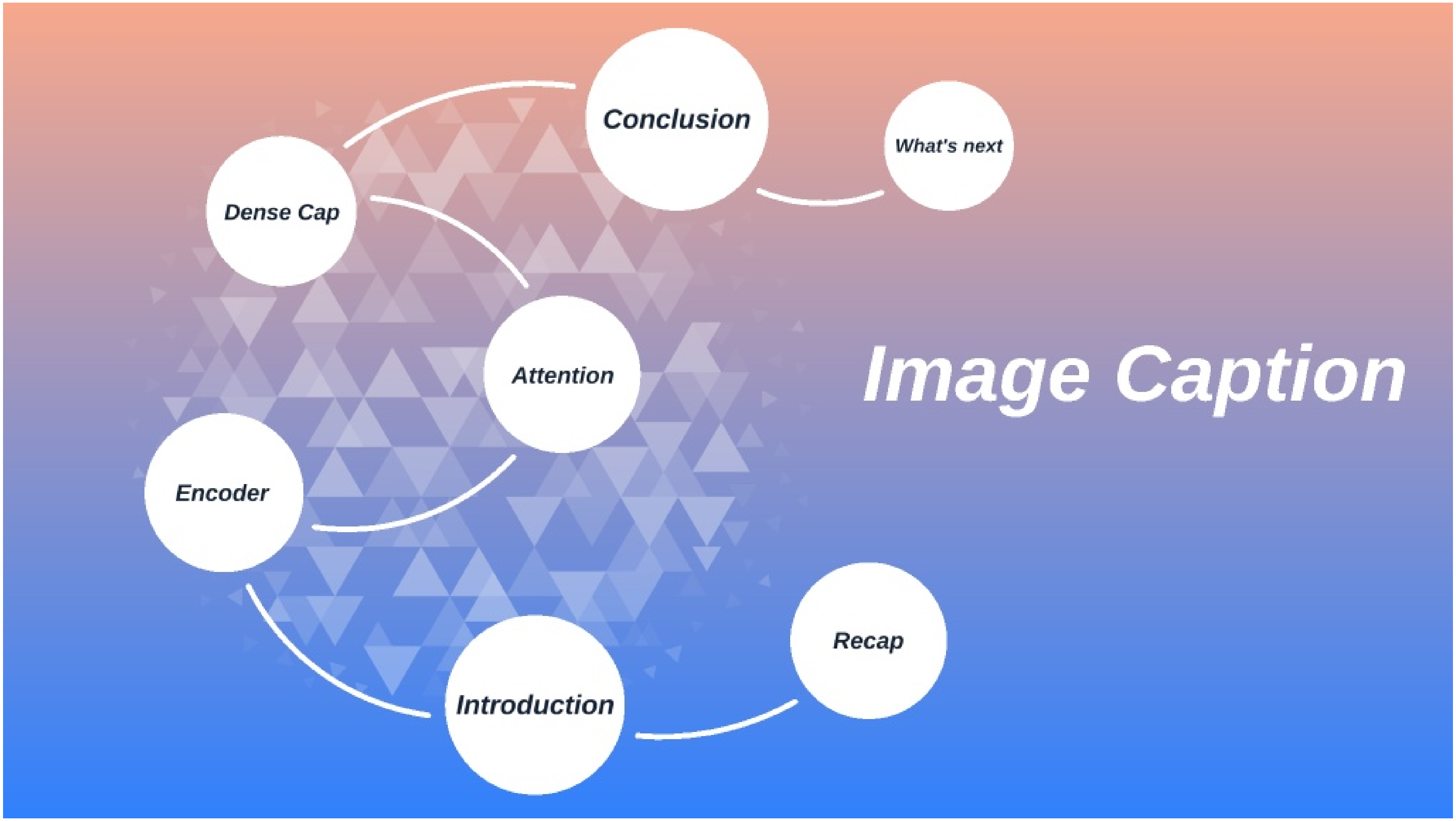


Image Caption

Recap

Introduction

Encoder

Attention

Dense Cap

Conclusion

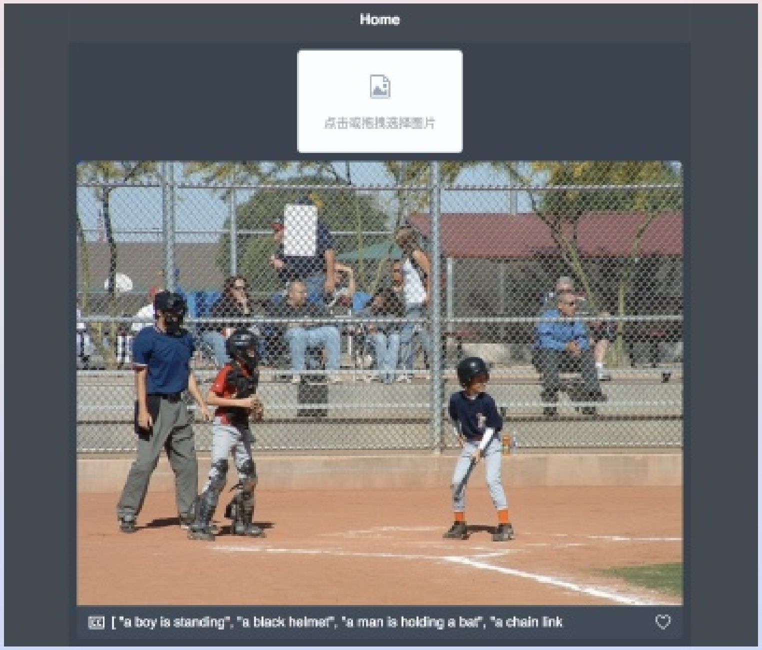
What's next

Recap

- 系统架构
- 基本Im2txt模型

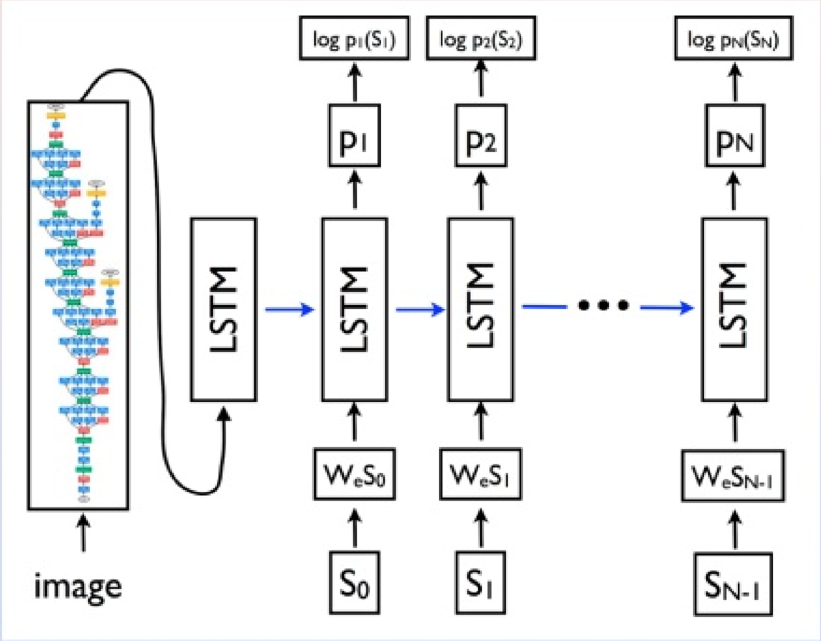
Model

Im2txt



["a boy is standing", "a black helmet", "a man is holding a bat", "a chain link

Model


log $p_1(S_1)$
log $p_2(S_2)$
log $p_N(S_N)$
p_1
p_2
p_N
LSTM
LSTM
LSTM
LSTM
W_eS_0
W_eS_1
W_eS_{N-1}
S_0
S_1
S_{N-1}
image

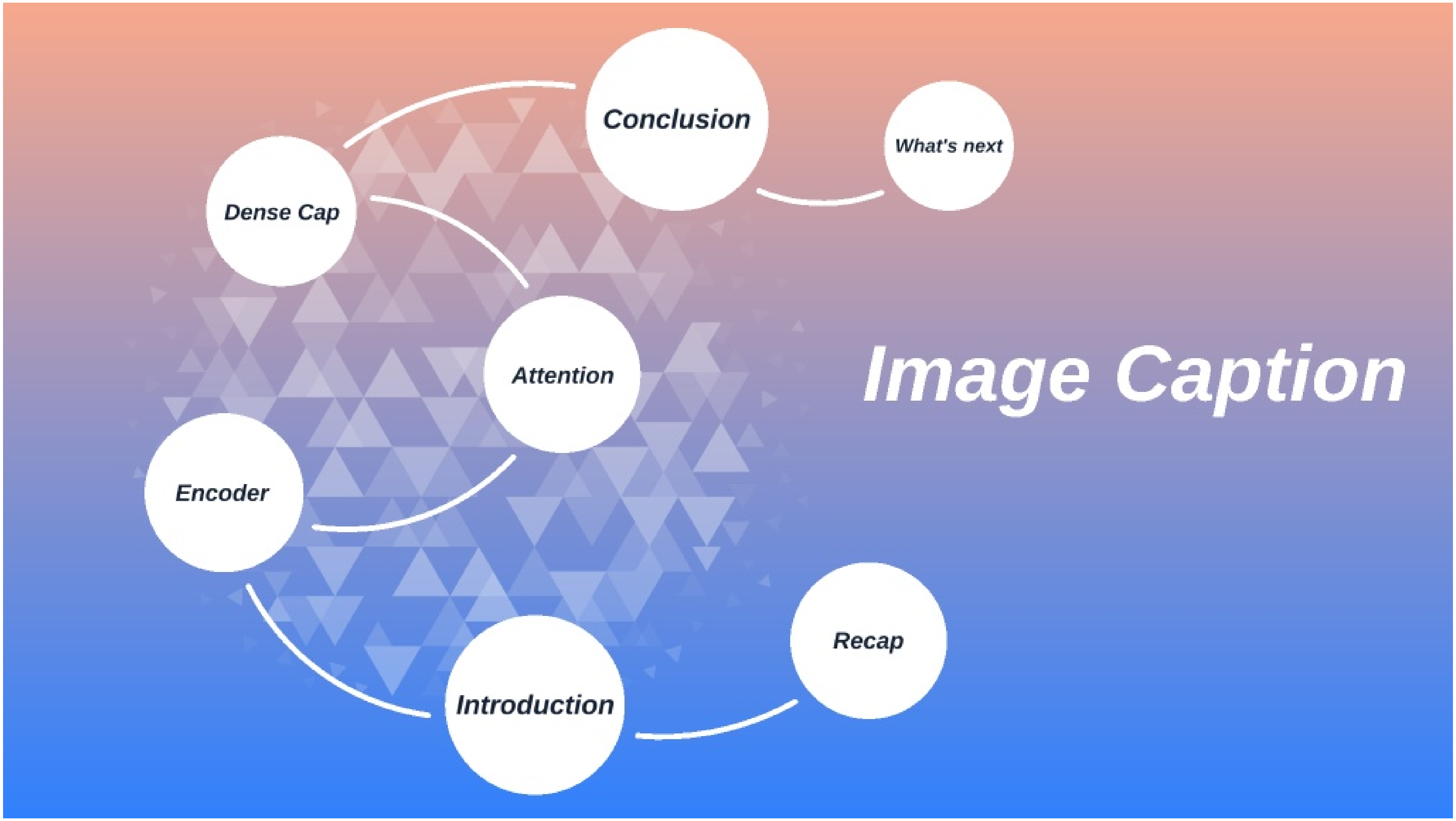


Image Caption

What's today

- Encoder
- Attention
- Dense Cap

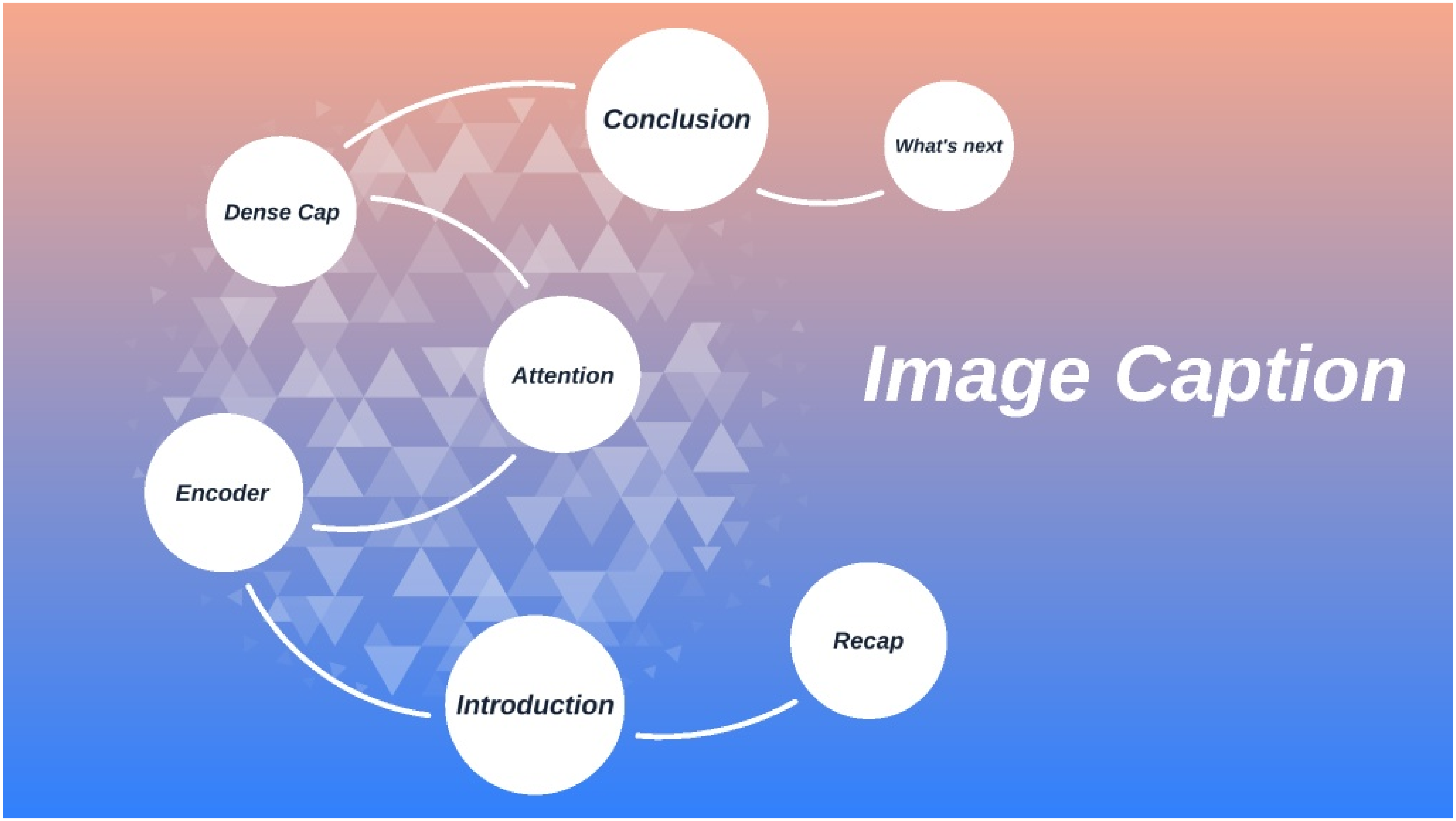


Image Caption

Conclusion

What's next

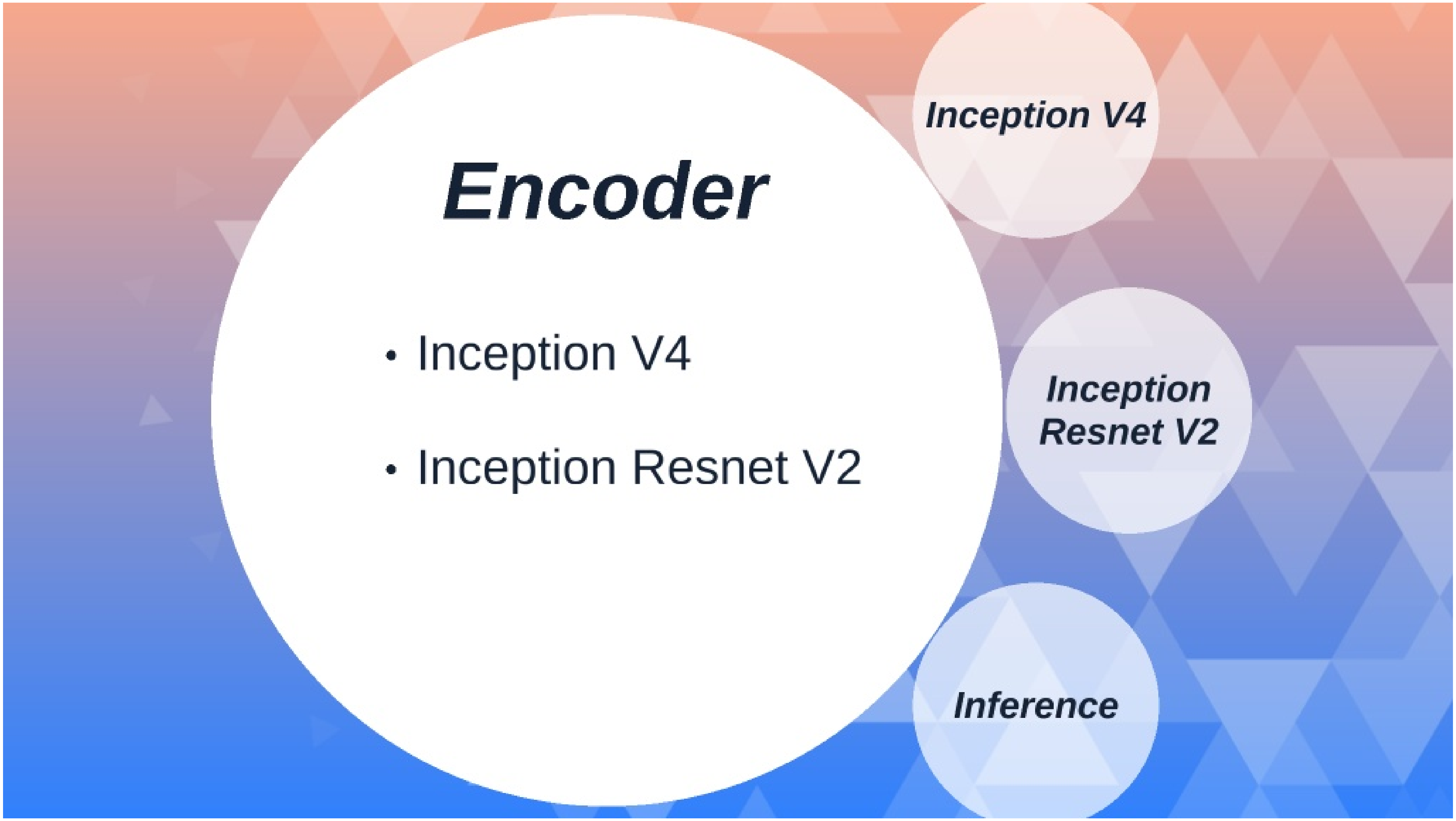
Dense Cap

Attention

Encoder

Introduction

Recap



Encoder

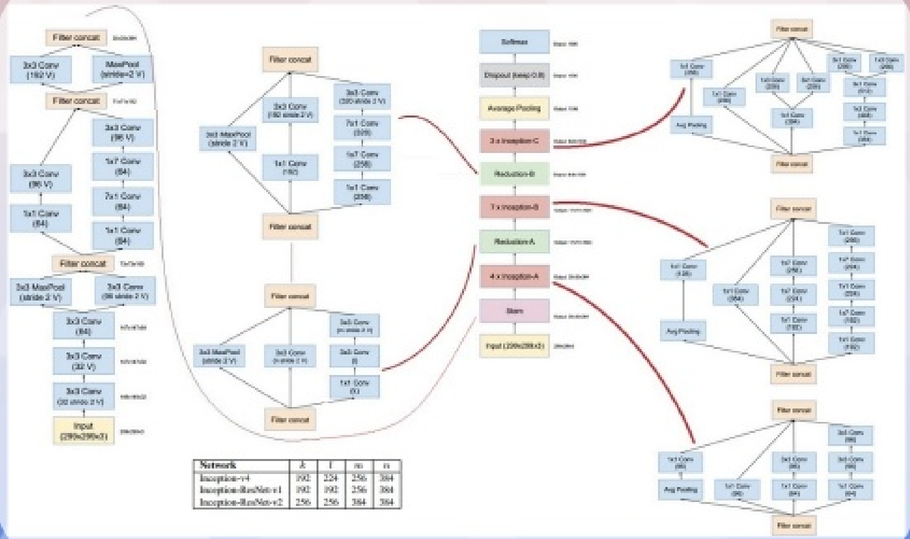
- Inception V4
- Inception Resnet V2

Inception V4

Inception Resnet V2

Inference

Inception V4

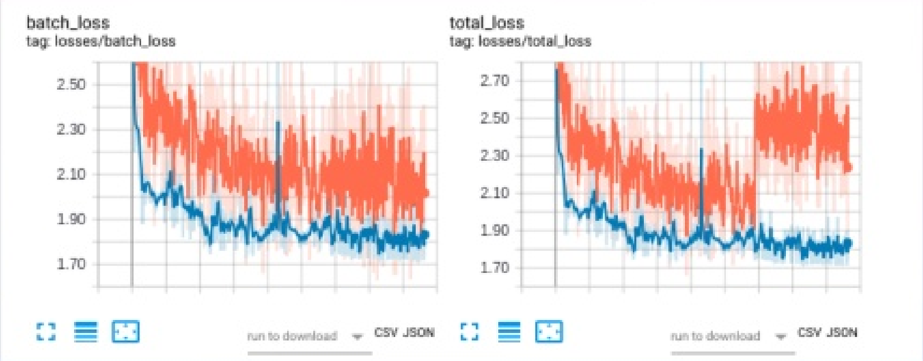


Network	k	l	m	n
Inception-v4	192	224	256	384
Inception-ResNet-v1	192	192	256	384
Inception-ResNet-v2	256	256	384	384

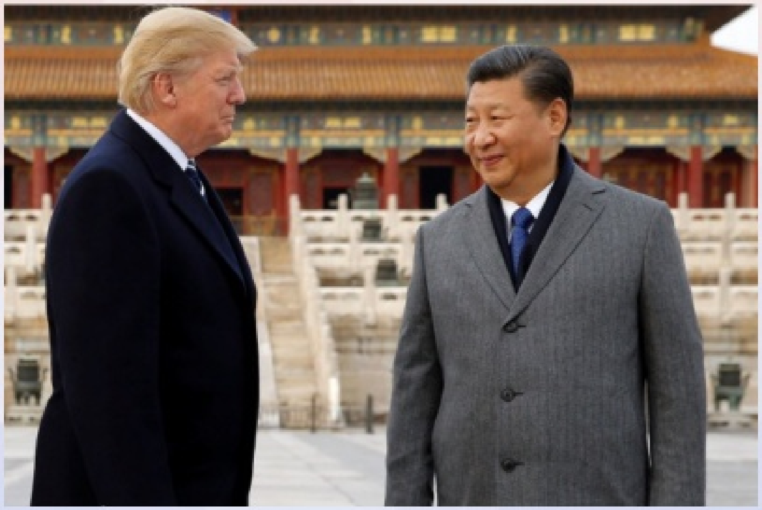
Current Stage

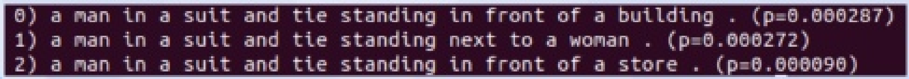
Inference

Training situation



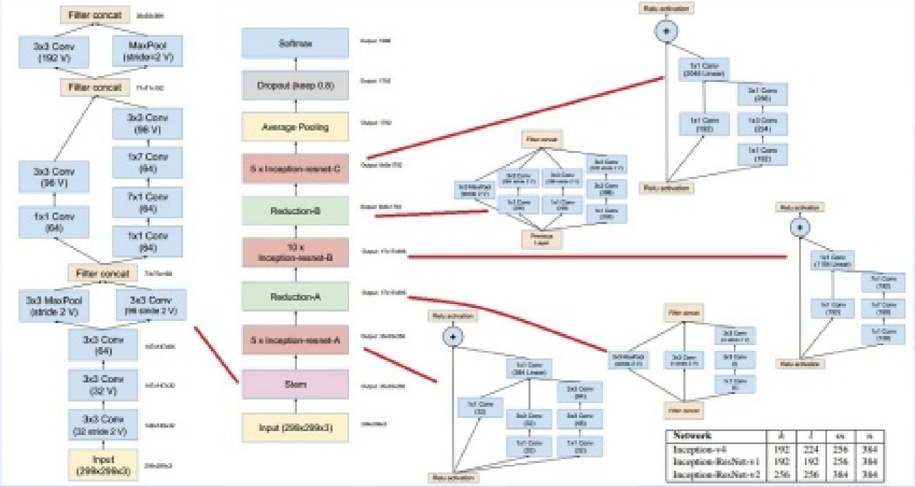
After 2 million steps,
loss on evaluation: 1.8





```
0) a man in a suit and tie standing in front of a building . (p=0.000287)
1) a man in a suit and tie standing next to a woman . (p=0.000272)
2) a man in a suit and tie standing in front of a store . (p=0.000090)
```

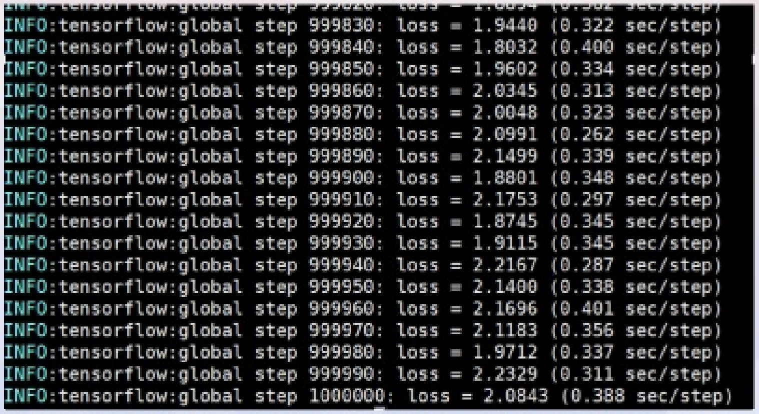
Inception Resnet V2



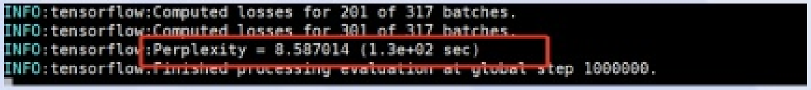
Network	k	l	m	n
Inception-v4	192	224	256	384
Inception-ResNet-v1	192	192	256	384
Inception-ResNet-v2	256	256	384	384

Current Stage

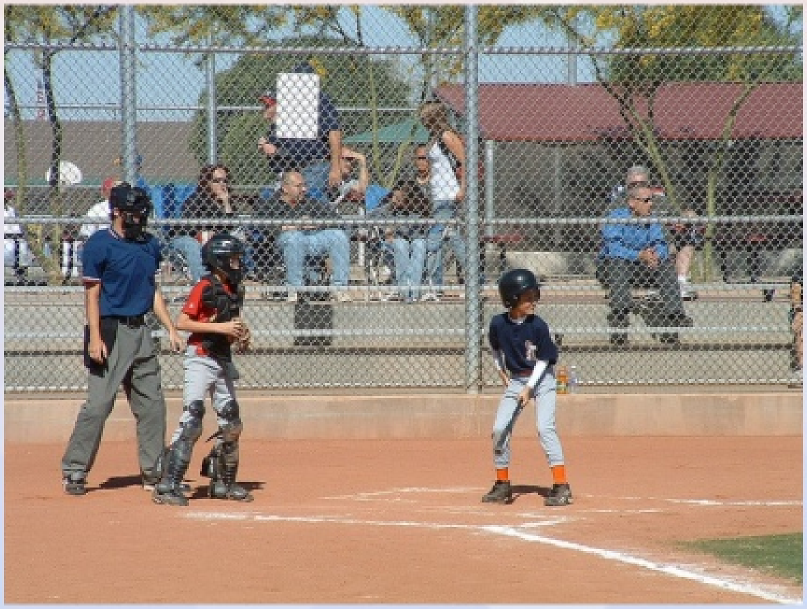
Training situation



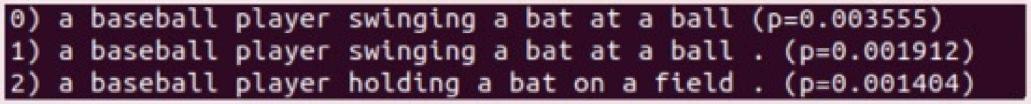
```
INFO:tensorflow:global step 999820: loss = 1.8894 (0.302 sec/step)
INFO:tensorflow:global step 999830: loss = 1.9440 (0.322 sec/step)
INFO:tensorflow:global step 999840: loss = 1.8032 (0.400 sec/step)
INFO:tensorflow:global step 999850: loss = 1.9602 (0.334 sec/step)
INFO:tensorflow:global step 999860: loss = 2.0345 (0.313 sec/step)
INFO:tensorflow:global step 999870: loss = 2.0048 (0.323 sec/step)
INFO:tensorflow:global step 999880: loss = 2.0991 (0.262 sec/step)
INFO:tensorflow:global step 999890: loss = 2.1499 (0.339 sec/step)
INFO:tensorflow:global step 999900: loss = 1.8801 (0.348 sec/step)
INFO:tensorflow:global step 999910: loss = 2.1753 (0.297 sec/step)
INFO:tensorflow:global step 999920: loss = 1.8745 (0.345 sec/step)
INFO:tensorflow:global step 999930: loss = 1.9115 (0.345 sec/step)
INFO:tensorflow:global step 999940: loss = 2.2167 (0.287 sec/step)
INFO:tensorflow:global step 999950: loss = 2.1400 (0.338 sec/step)
INFO:tensorflow:global step 999960: loss = 2.1696 (0.401 sec/step)
INFO:tensorflow:global step 999970: loss = 2.1183 (0.356 sec/step)
INFO:tensorflow:global step 999980: loss = 1.9712 (0.337 sec/step)
INFO:tensorflow:global step 999990: loss = 2.2329 (0.311 sec/step)
INFO:tensorflow:global step 1000000: loss = 2.0843 (0.388 sec/step)
```



```
INFO:tensorflow:Computed losses for 201 of 317 batches.
INFO:tensorflow:Computed losses for 301 of 317 batches.
INFO:tensorflow:Perplexity = 8.587014 (1.3e+02 sec)
INFO:tensorflow:Finished processing evaluation at global step 1000000.
```

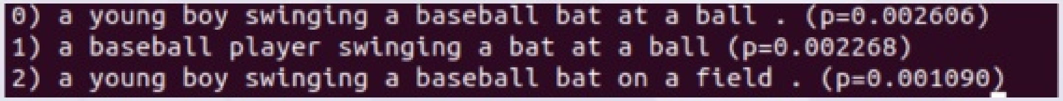


Inception V3:



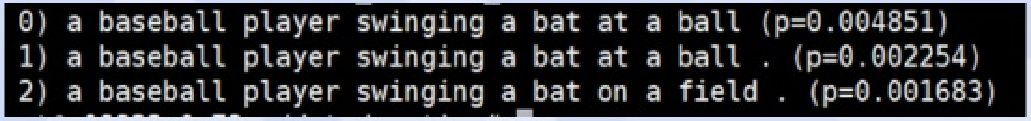
```
0) a baseball player swinging a bat at a ball (p=0.003555)
1) a baseball player swinging a bat at a ball . (p=0.001912)
2) a baseball player holding a bat on a field . (p=0.001404)
```

Inception V4:



```
0) a young boy swinging a baseball bat at a ball . (p=0.002606)
1) a baseball player swinging a bat at a ball (p=0.002268)
2) a young boy swinging a baseball bat on a field . (p=0.001090)
```

Inception Resnet V2:



```
0) a baseball player swinging a bat at a ball (p=0.004851)
1) a baseball player swinging a bat at a ball . (p=0.002254)
2) a baseball player swinging a bat on a field . (p=0.001683)
```

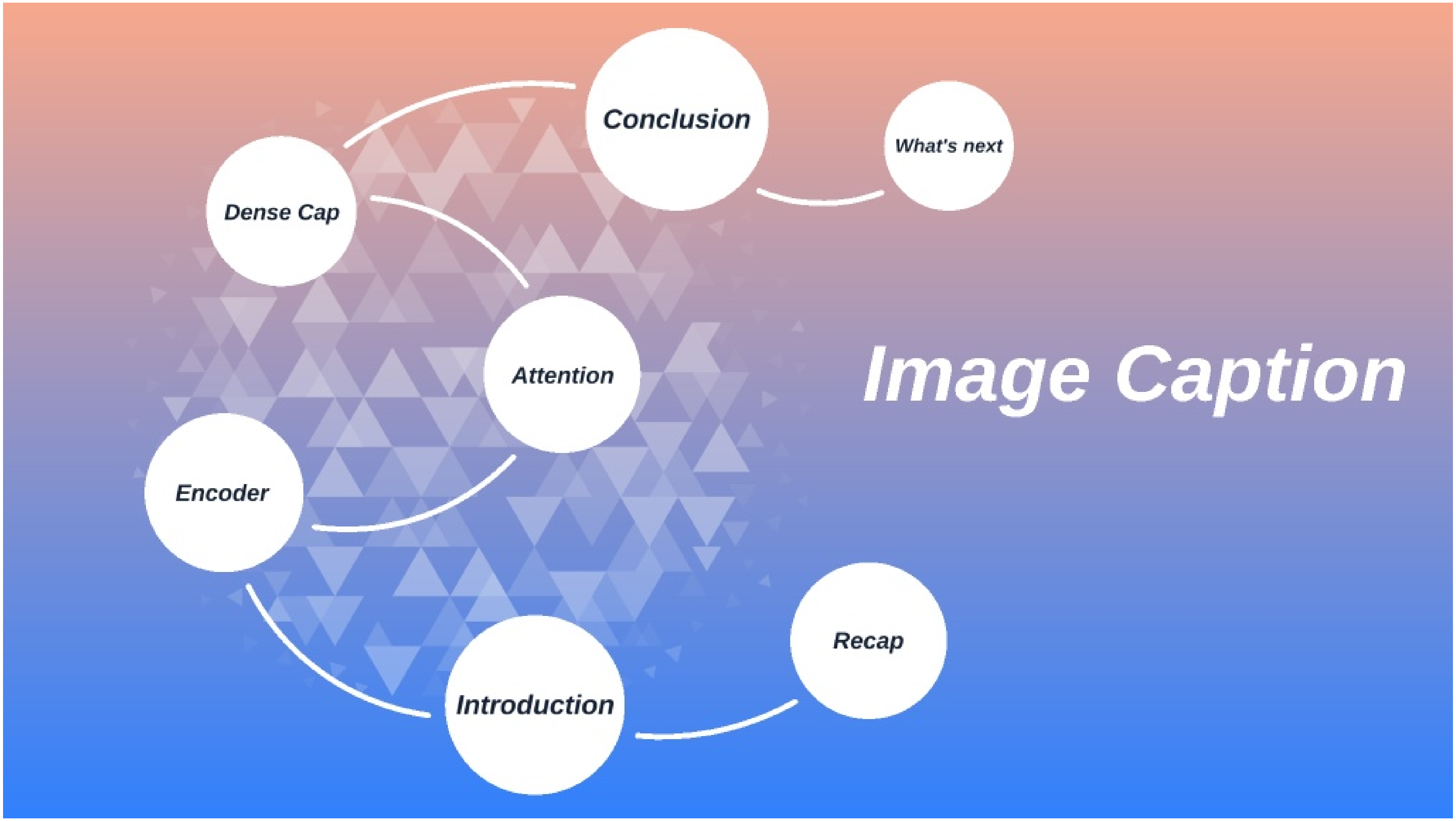


Image Caption

Conclusion

What's next

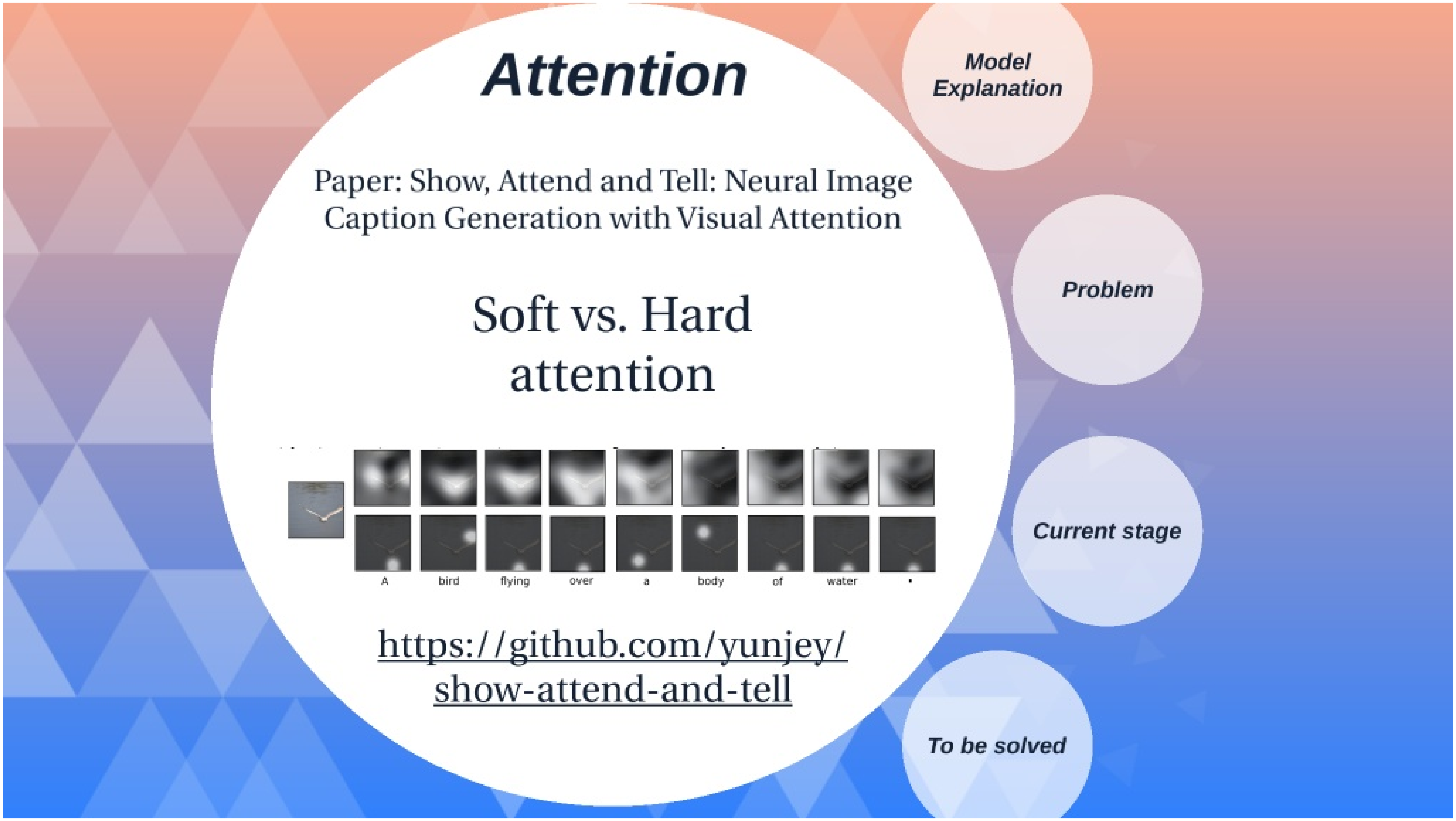
Dense Cap

Attention

Encoder

Introduction

Recap



Attention

Paper: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Soft vs. Hard attention

https://github.com/yunjey/show-attend-and-tell

Model Explanation

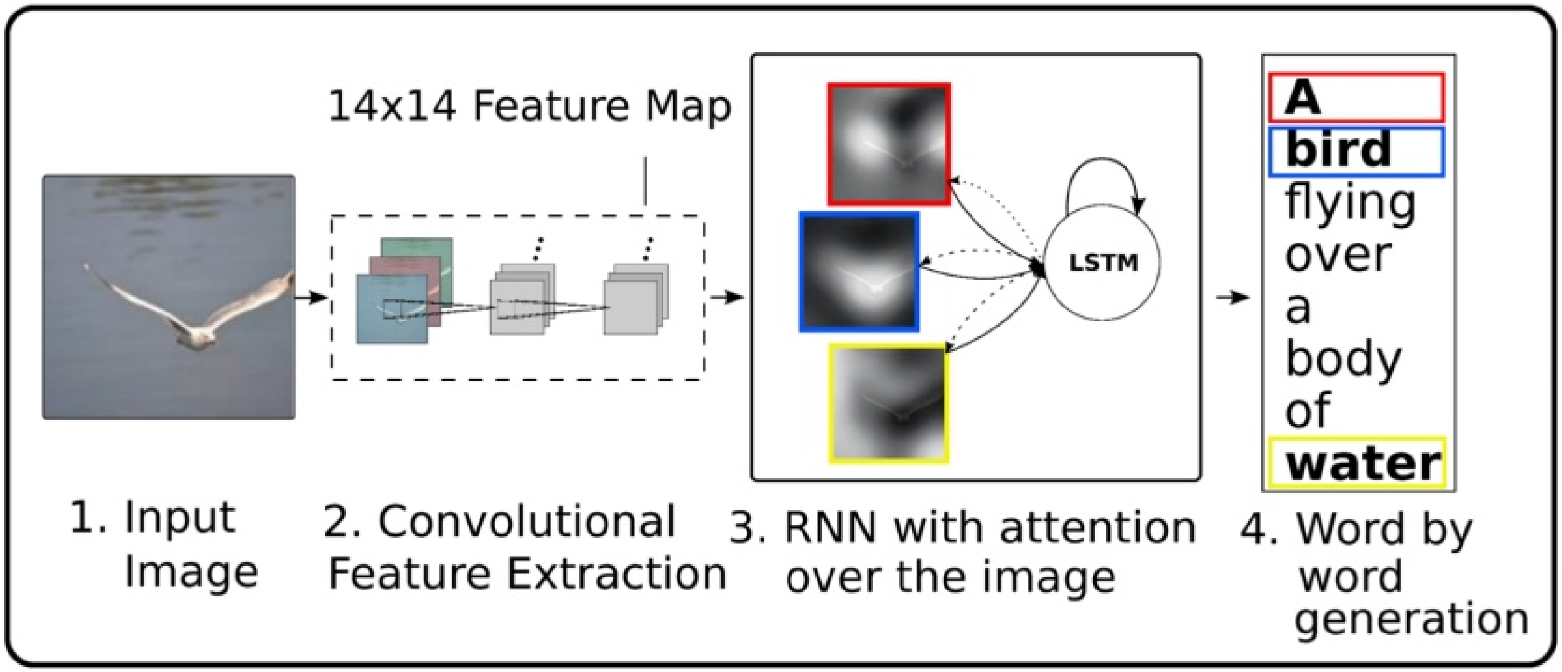
Problem

Current stage

To be solved

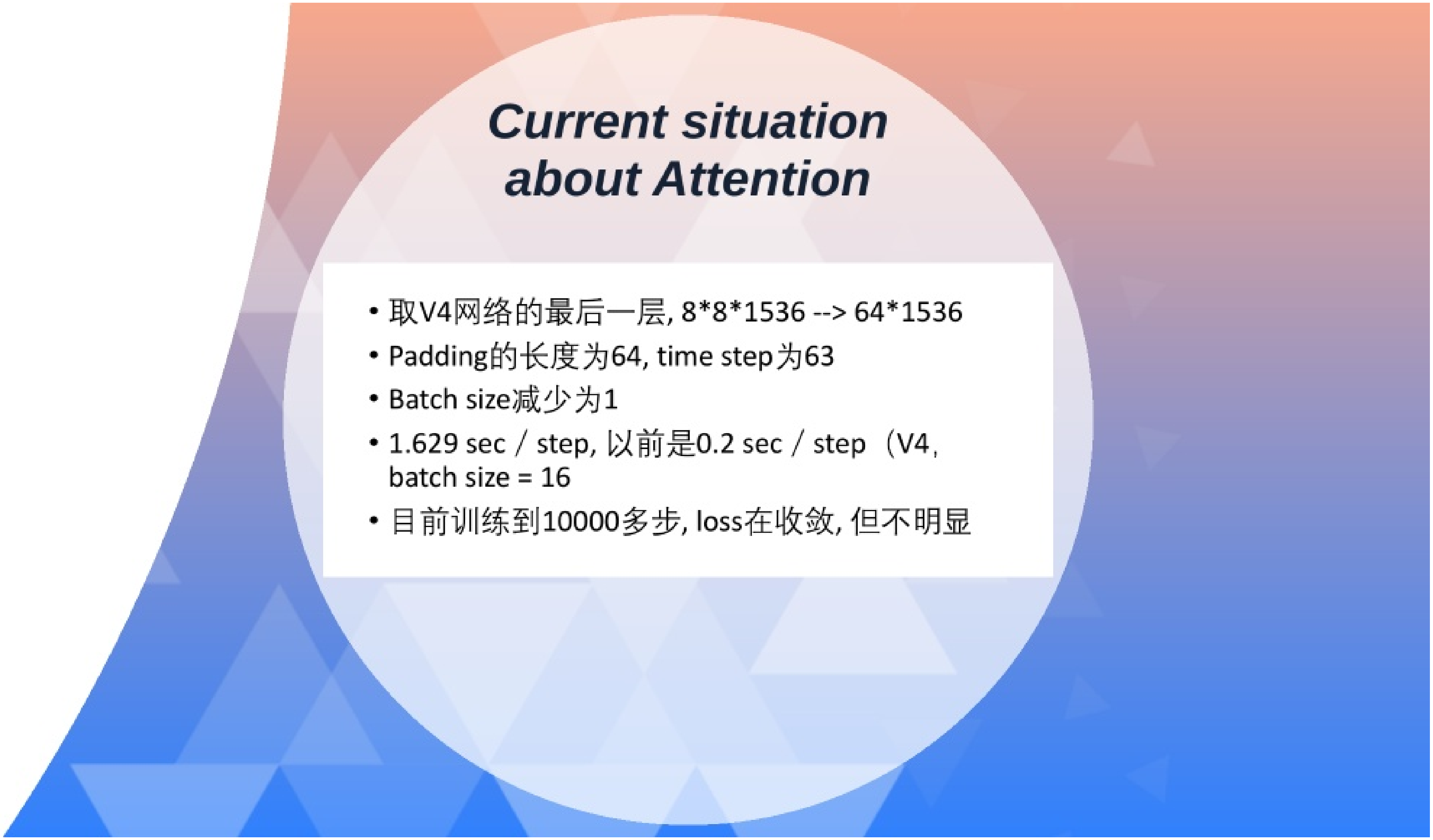
Attention layer in Im2txt

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (ICML 2015)


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

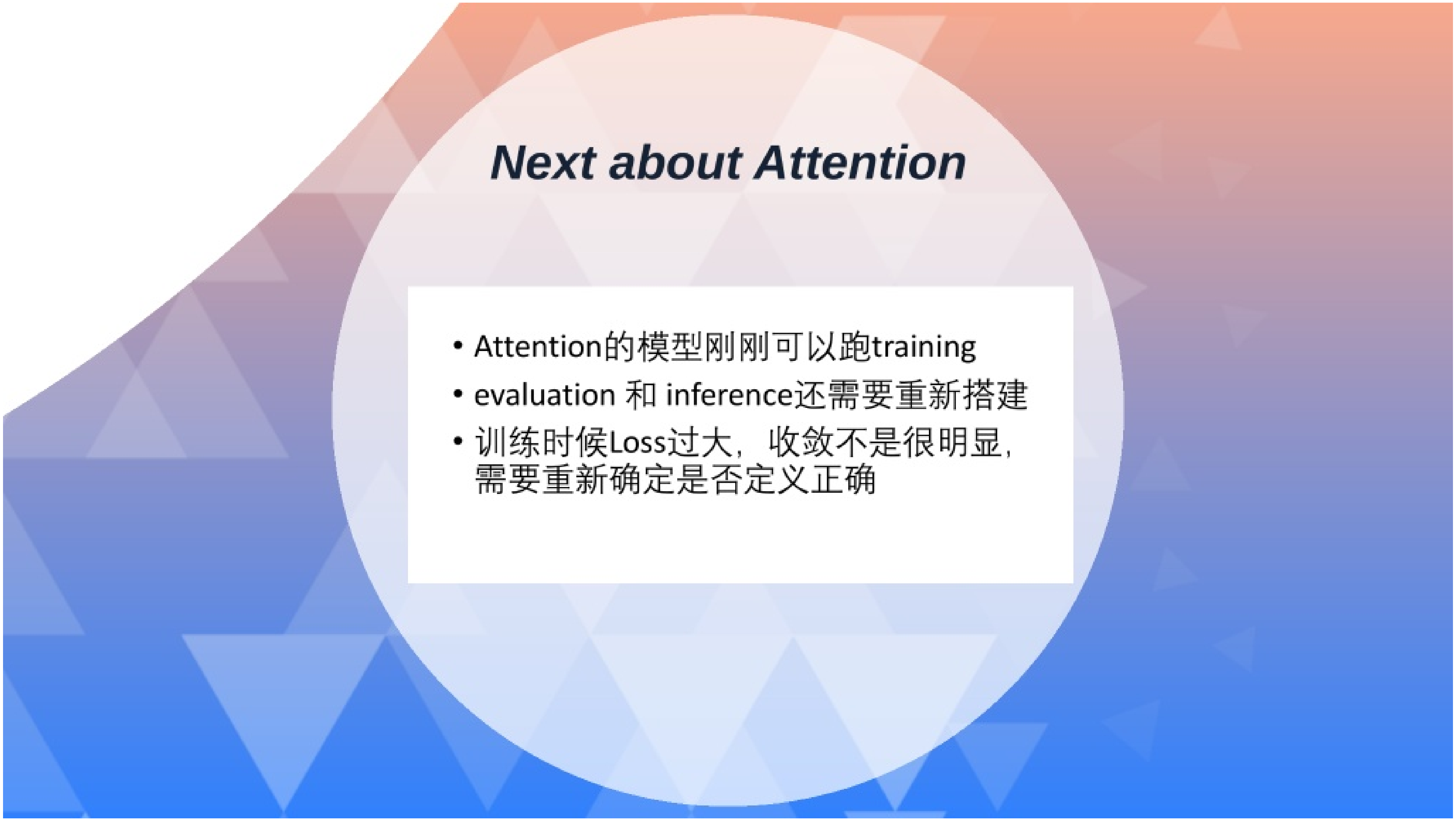
Problems about Attention

- Tensorflow下无可用的attention wrapper, 需要手动创建attention layer
- CNN不再用embedding, 而是取最后几层的feature, 维度大
- 基础的im2txt模型用了dynamic_rnn, 然而attention layer无法用高级api封装, 需要设定padding的长度从而指定time step



Current situation about Attention

- 取V4网络的最后一层, 8*8*1536 --> 64*1536
- Padding的长度为64, time step为63
- Batch size减少为1
- 1.629 sec / step, 以前是0.2 sec / step（V4，batch size = 16
- 目前训练到10000多步, loss在收敛, 但不明显



Next about Attention

- Attention的模型刚刚可以跑training
- evaluation 和 inference还需要重新搭建
- 训练时候Loss过大，收敛不是很明显，需要重新确定是否定义正确

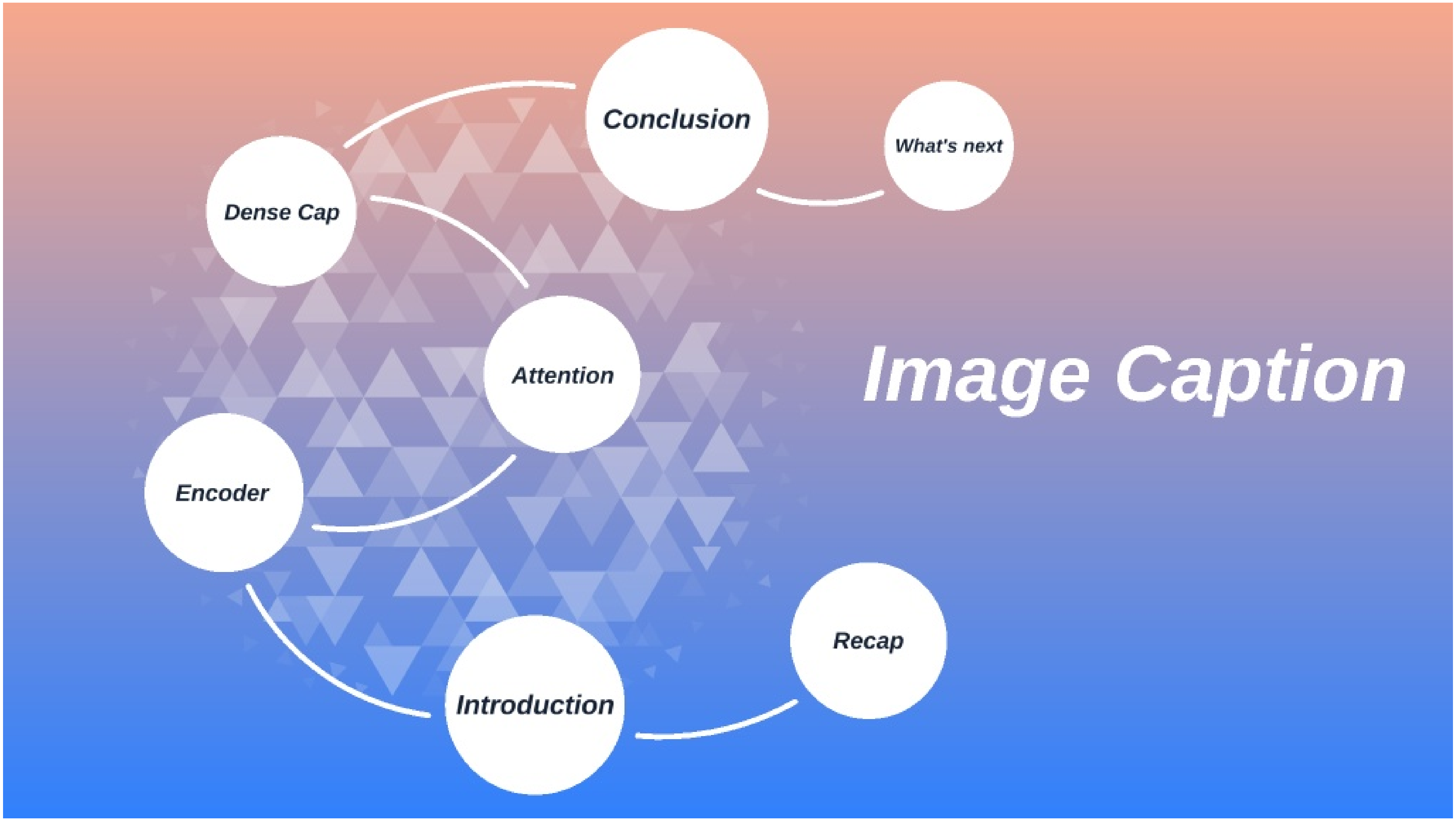


Image Caption

Conclusion

What's next

Dense Cap

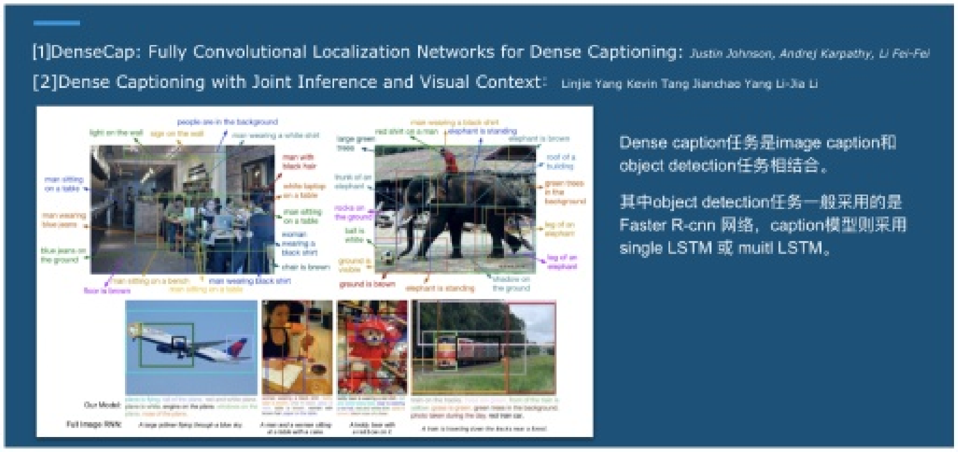
Attention

Encoder

Introduction

Recap

Dense Cap



[1]DenseCap: Fully Convolutional Localization Networks for Dense Captioning: *Justin Johnson, Andrej Karpathy, Li Fei-Fei*

[2]Dense Captioning with Joint Inference and Visual Context: Linjie Yang Kevin Tang Jianchao Yang Li-Jia Li

Dense caption任务是image caption和object detection任务相结合。

其中object detection任务一般采用的是Faster R-cnn 网络，caption模型则采用single LSTM 或 muitl LSTM。

Model

Faster R CNN

Loss

Data

Result

Model

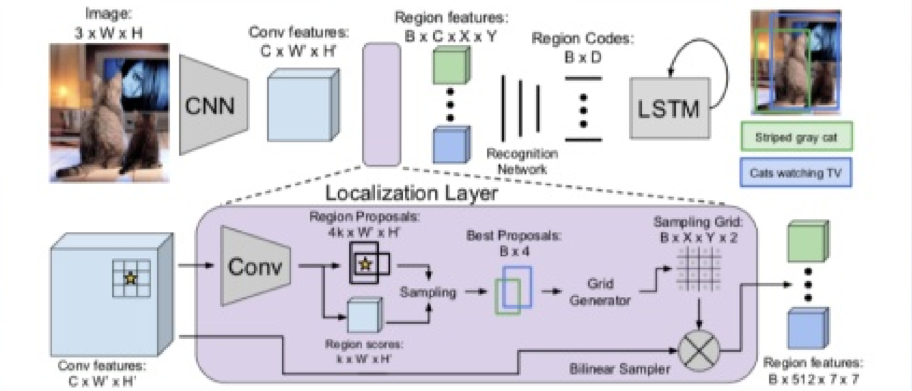
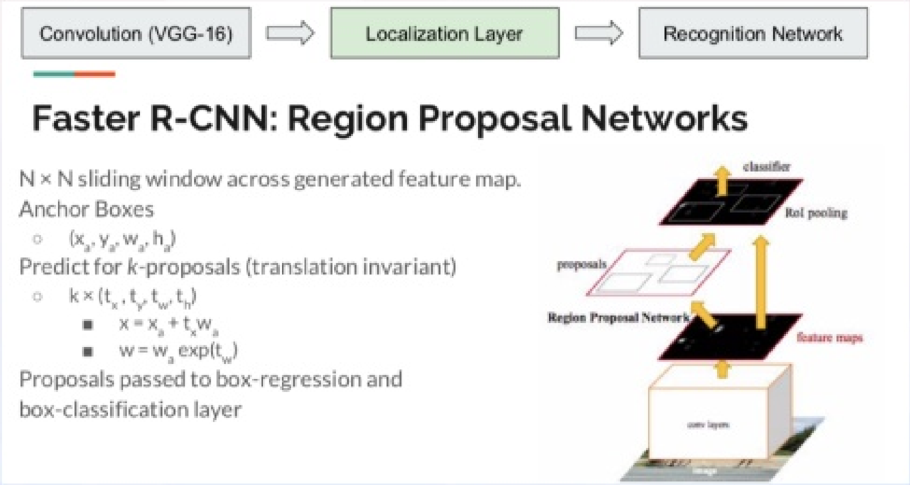


Figure 2. Model overview. An input image is first processed a CNN. The Localization Layer proposes regions and smoothly extracts a batch of corresponding activations using bilinear interpolation. These regions are processed with a fully-connected recognition network and described with an RNN language model. The model is trained end-to-end with gradient descent.

Faster R-CNN



Convolution (VGG-16) ⇨ Localization Layer ⇨ Recognition Network

Faster R-CNN: Region Proposal Networks

N × N sliding window across generated feature map.

Anchor Boxes

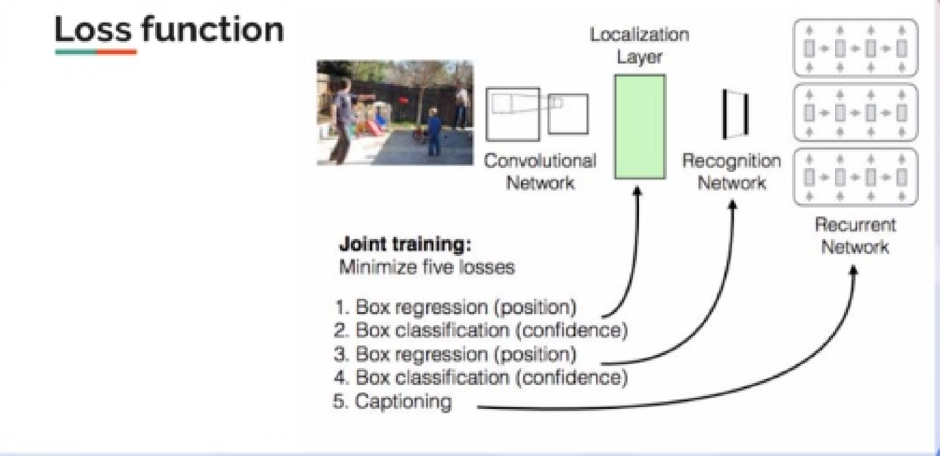
- (x_a, y_a, w_a, h_a)

Predict for *k*-proposals (translation invariant)

- $k \times (t_x, t_y, t_w, t_h)$
    - $x = x_a + t_x w_a$
    - $w = w_a \exp(t_w)$

Proposals passed to box-regression and box-classification layer

Loss Function



Loss function

Joint training:
Minimize five losses

1. Box regression (position)
2. Box classification (confidence)
3. Box regression (position)
4. Box classification (confidence)
5. Captioning

Dataset

Li Fei-Fei团队设计了这样一个数据集：它不但包括了图像本身，更包括了图像内对象之间的关系等众多数据（包括objects、attributes、relationship等），并希望通过这些数据能够推动"认知"这一问题在CV领域的发展。

Visual Genome一共包括了 **108K张图片**，平均每张图片内包含了 **35个object**， 和 **26个attributes**，以及 **21对object之间的relationship pair**。本数据集的图片取自MS COCO 和 YFCC100M。

除此之外，作者们还将其中所有的object、attributes、relationships和在region descriptions与question answer pairs中的名词短语都映射到了WordNet synset上。从而让打通了从CV到Knowledge乃至NLP之间的连接通道。

在densecap实验中数据集的预处理包括：对于描述的的内容，去掉了类似于"there is…"和"this seems to be a"这一类的referring phrases。为了效率去除了大于10个单词的注释，另外还有注释个数小于20或者大于50的图片。最终留下的有87398张图， validation sets和test sets各分得5000张图。

实验中数据集划分：

train set ： 77398 examples.
val set ： 5000 examples.
test set ： 5000 examples.

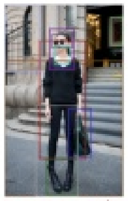
Current Result

我们对im2txt模型改进阶段的代码基于2篇论文：'DenseCap: Fully Convolutional Localization Networks for Dense Captioning '& 'Dense Captioning with Joint Inference and Visual Context'
根据官方代码进行修改。

实验中提取基础特征CNN我们采用在imagenet中进行了预训练的Resnet-50网络，Faster r-cnn 部分则采用了在COCO数据集上的预训练模型，最后我们的网络在Visual Genome数据集上进行训练。

由于模型中由多个不同的结构构成，所以模型的参数较多，并且在调参过程中分了几个不同的阶段对参数进行调试，较im2txt调参过程更加复杂。（右图为训练50K的模型效果。）

a man wearing a blue shirt. a woman wearing sunglasses. a pair of sunglasses. the head of a man. the man is wearing a black shirt. a white shirt on a woman. a woman wearing sunglasses. a yellow sign on the wall. the hand of a person. the shirt is black. a man in a black shirt. a black chair in the background. the woman has blonde hair. a building in the background. the hand of a man. people in the background. the arm of a man.

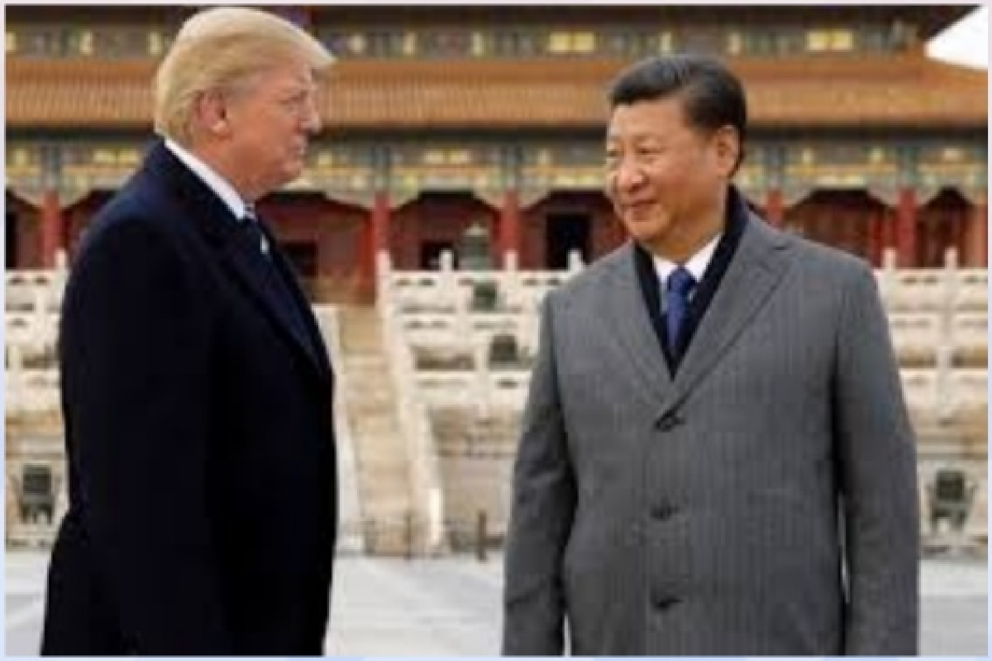


实验效果

a woman wearing a black jacket. the bag is green. black sunglasses on a woman. a pair of black boots. a woman walking on the sidewalk. a woman wearing sunglasses. the woman is wearing black pants. picture on the wall. black metal railing on the side of the building. the woman is wearing a necklace.

Inference

Inference



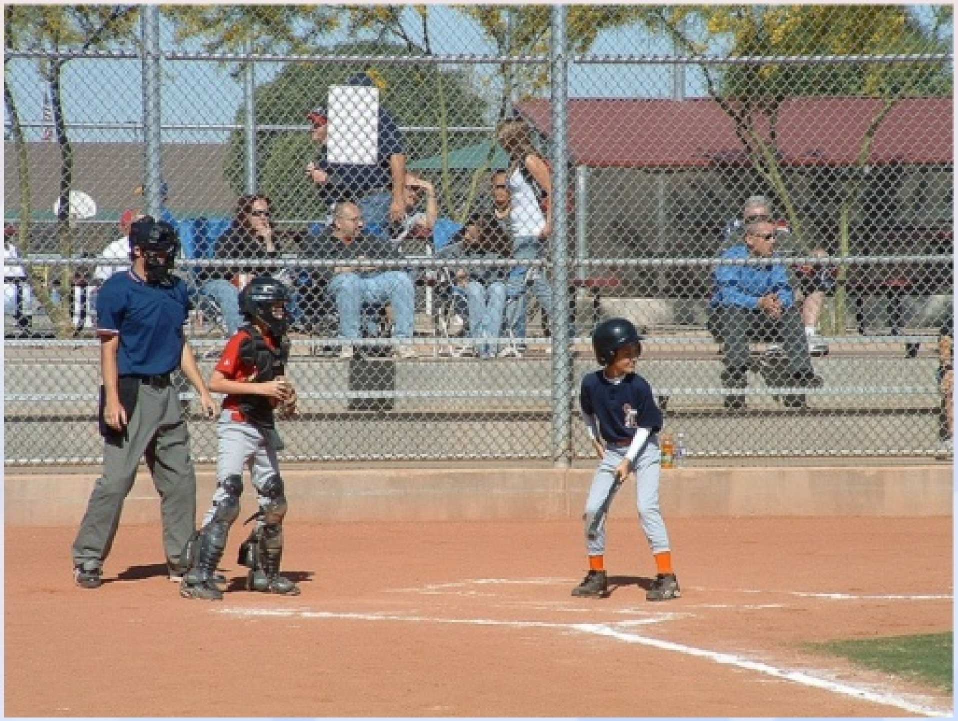
V4:

"a man in a suit and tie standing in front of a building"
"a man in a suit and tie standing next to a woman"
"a man in a suit and tie standing in front of a store"

DenseCap:

"a man wearing a black jacket"
"a man in a black jacket"
"the mans head is white"
"the head of a man"
"the mans shirt is black"
"the mans hair is black"
"a white building"

Inference



V3:
a baseball player swinging a bat at a ball

V4:
a young boy swinging a baseball bat at a ball

Resnet V2:
a baseball player swing a bat at a ball

DenseCap:
"boy is standing"
"a black helmet"
"a man is holding a bat"
"a chain link fence"
"a person sitting on a bench"
"a black helmet"
"the mans shoes are black"

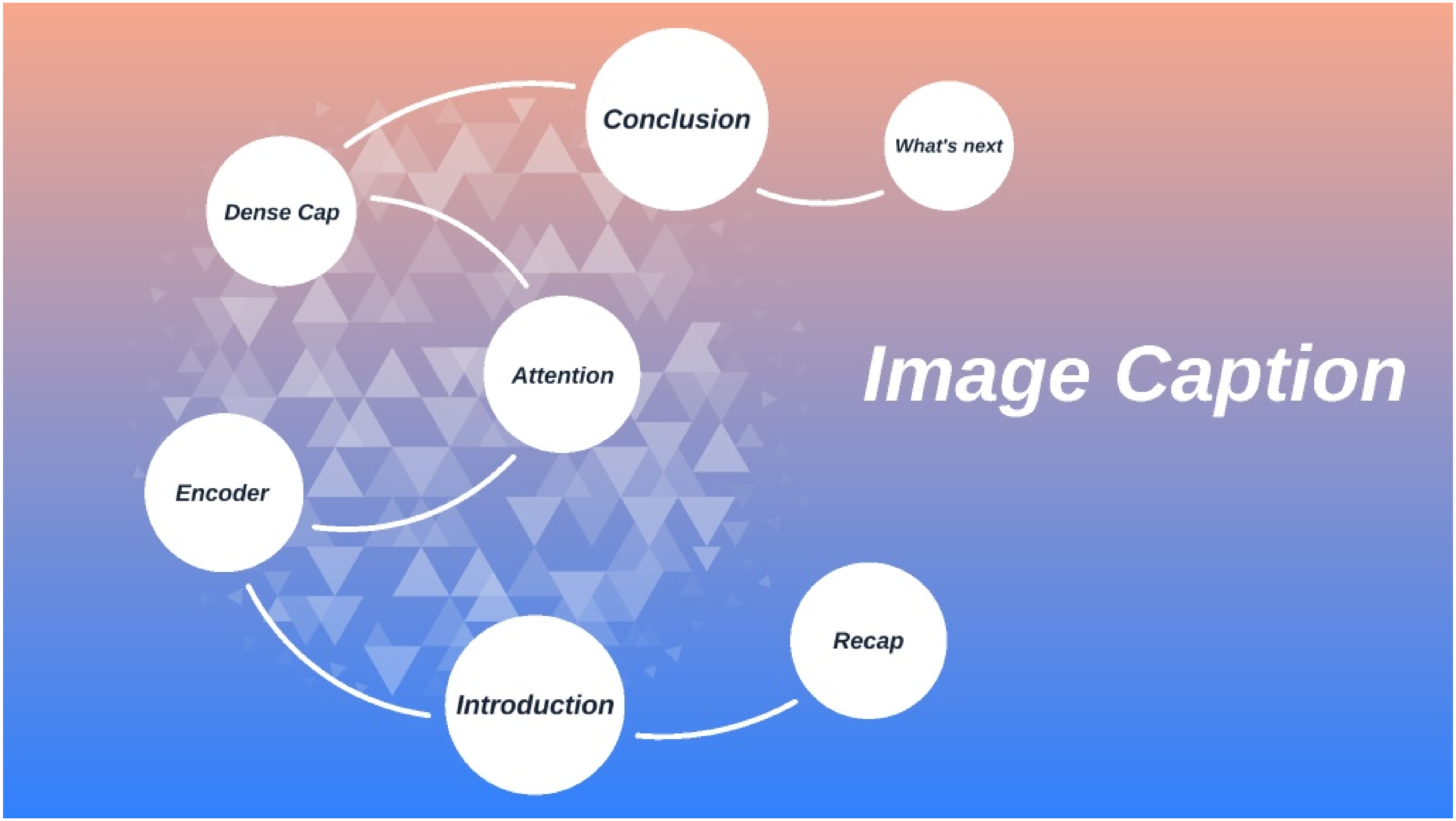


Image Caption

Conclusion

What's next

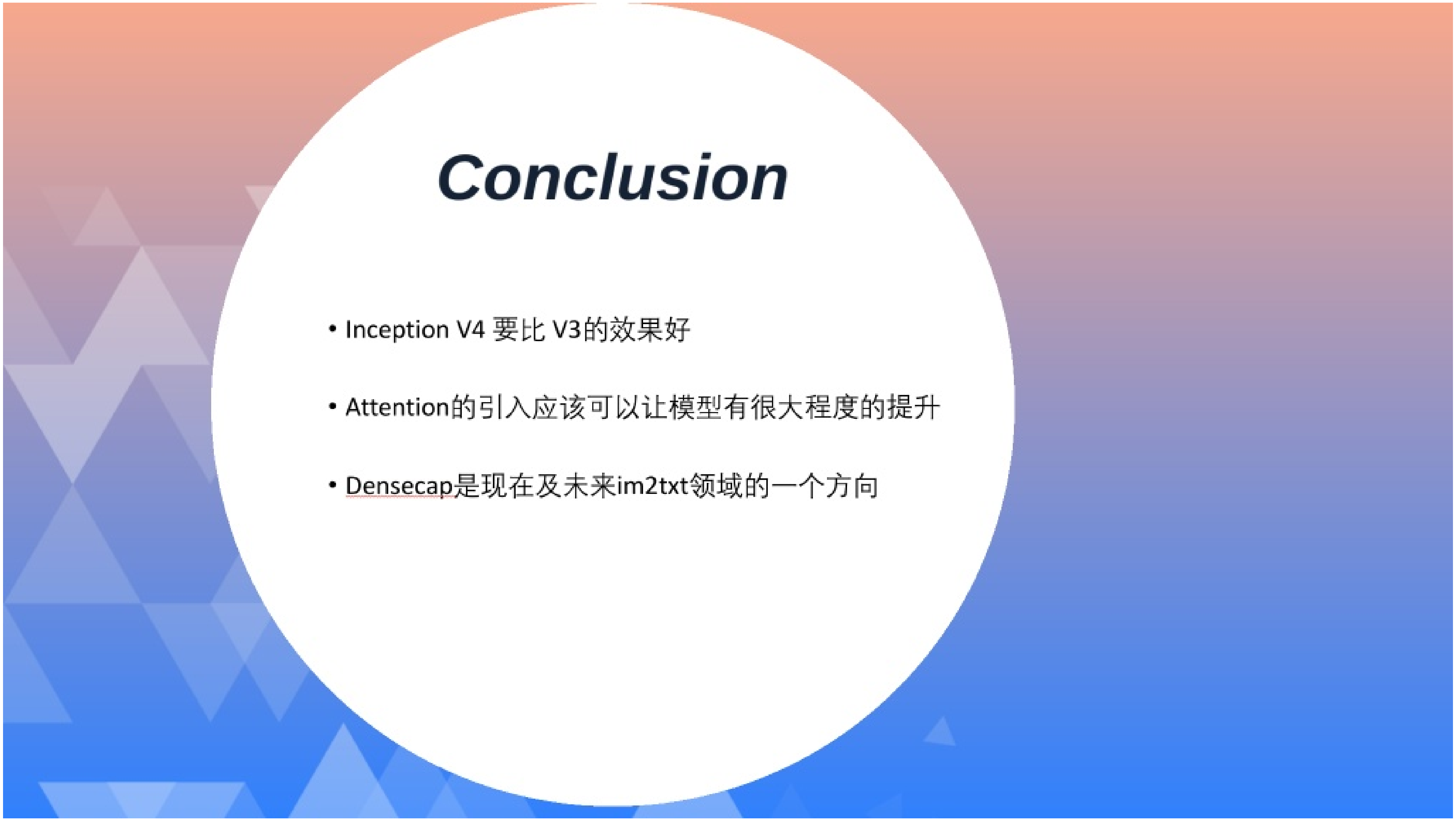
Dense Cap

Attention

Encoder

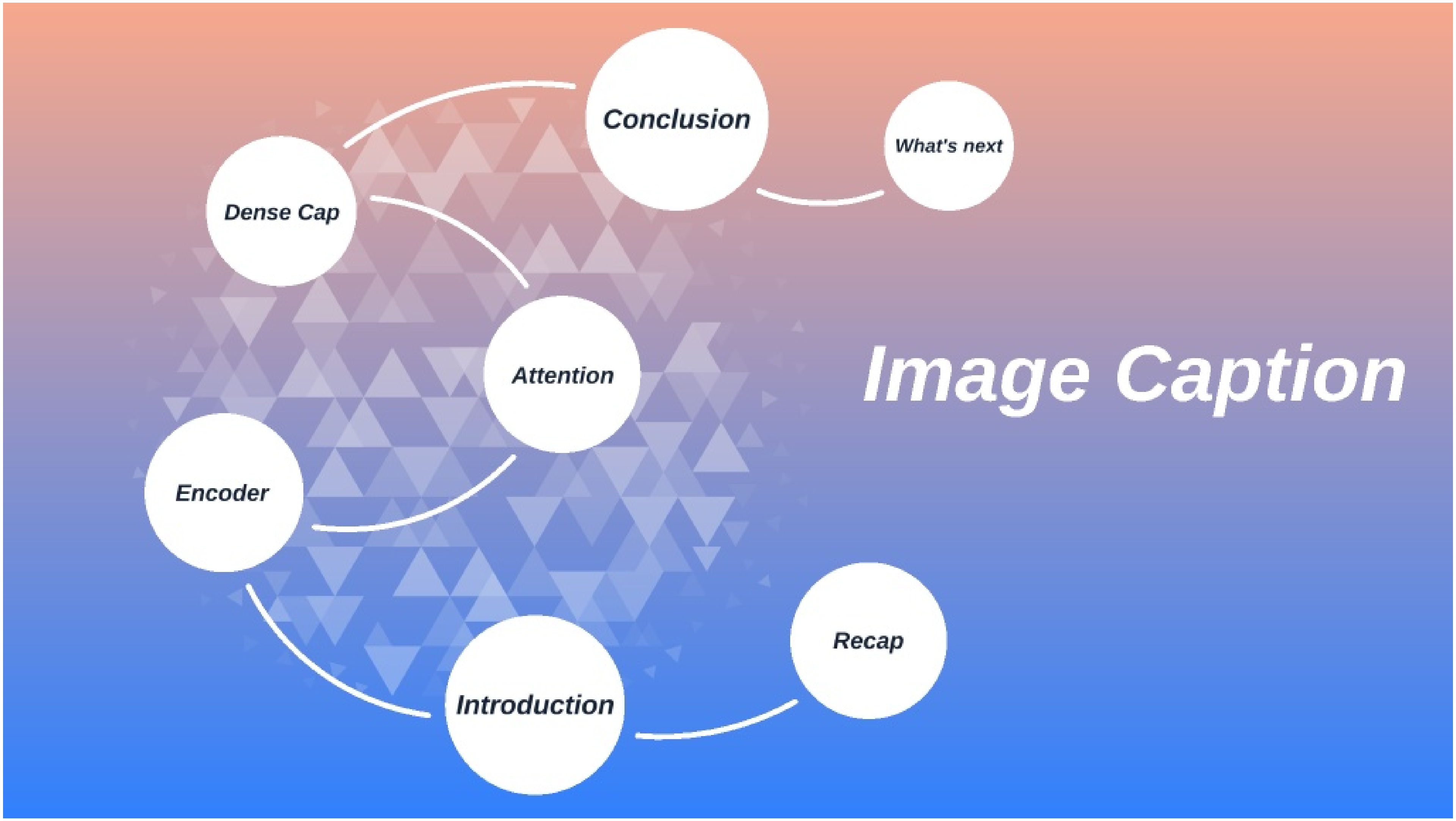
Introduction

Recap



Conclusion

- Inception V4 要比 V3的效果好
- Attention的引入应该可以让模型有很大程度的提升
- Densecap是现在及未来im2txt领域的一个方向



Conclusion

What's next

Dense Cap

Attention

Image Caption

Encoder

Recap

Introduction

What's next

- 确保Attention模型的正确
- 完善attention模型的evaluation 和 inference
- 尝试把attention加入dense cap里

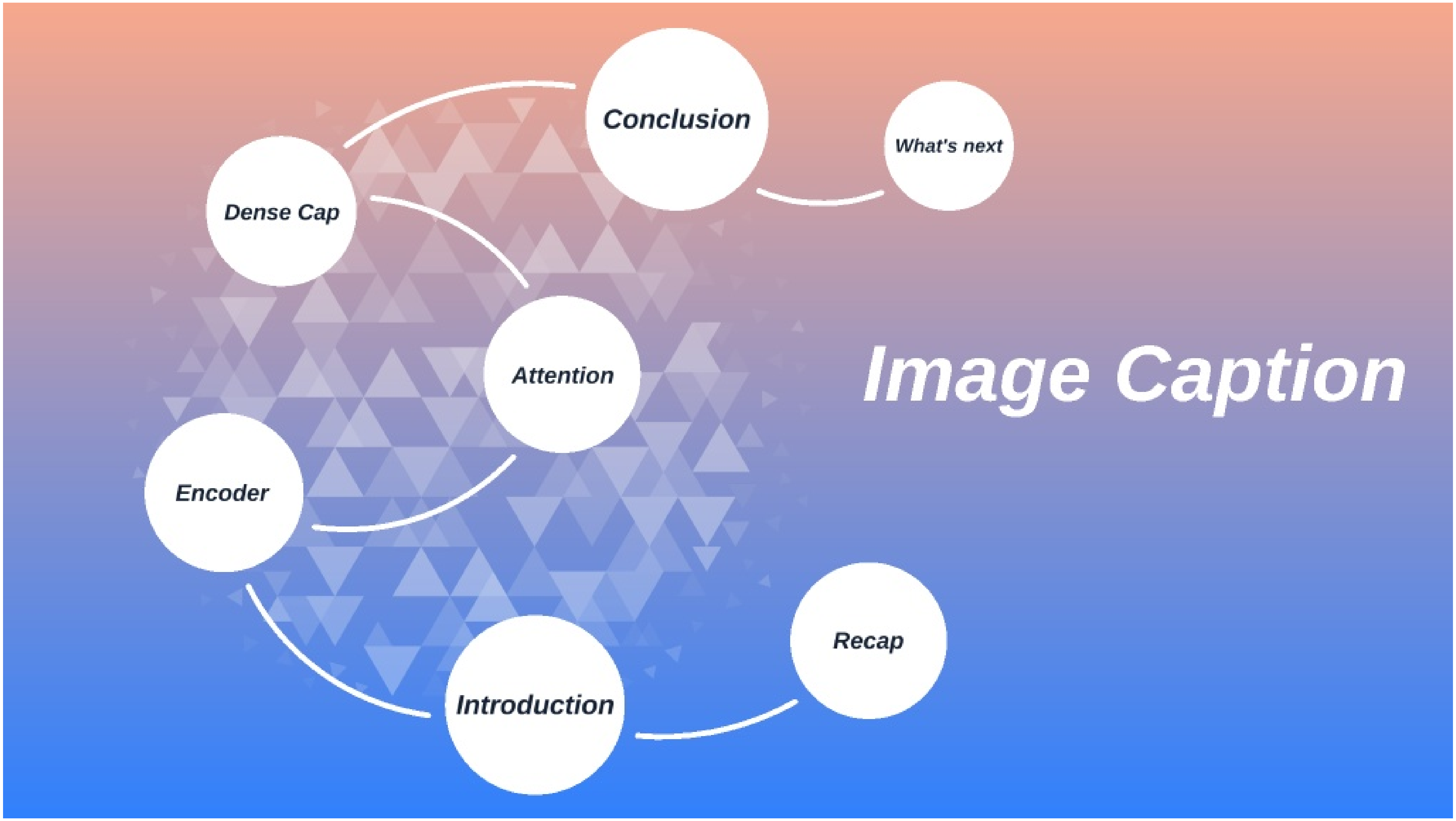


Image Caption

Conclusion

What's next

Dense Cap

Attention

Encoder

Introduction

Recap