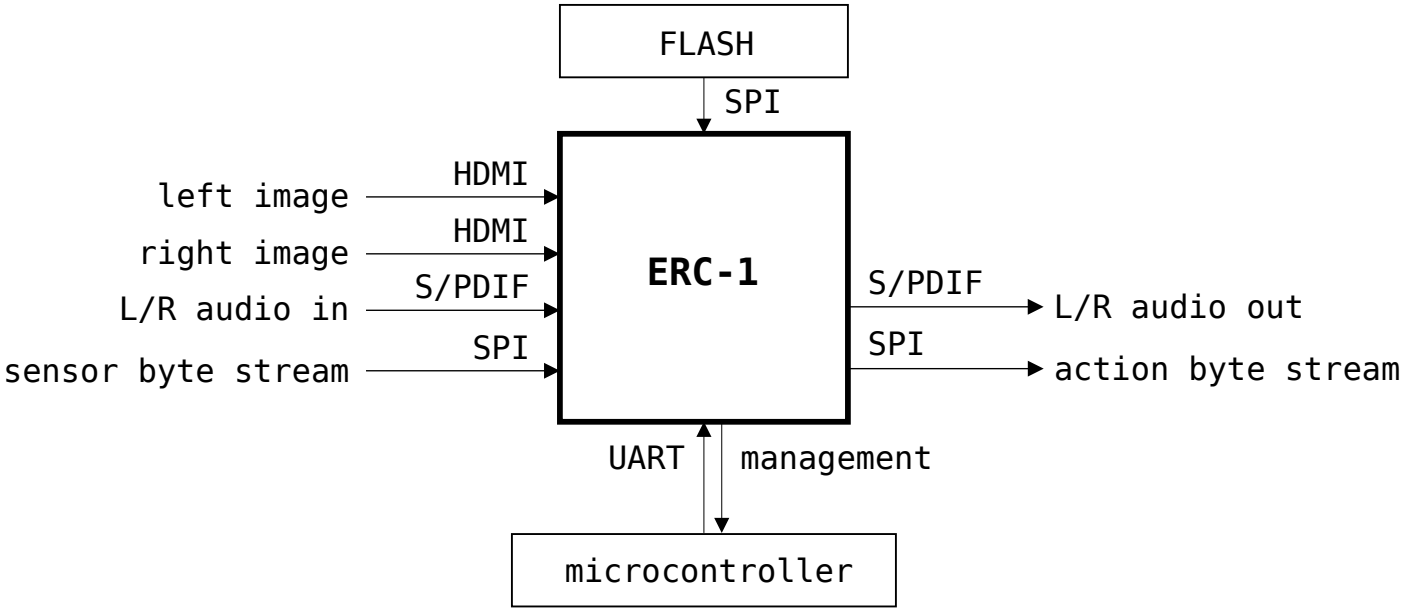
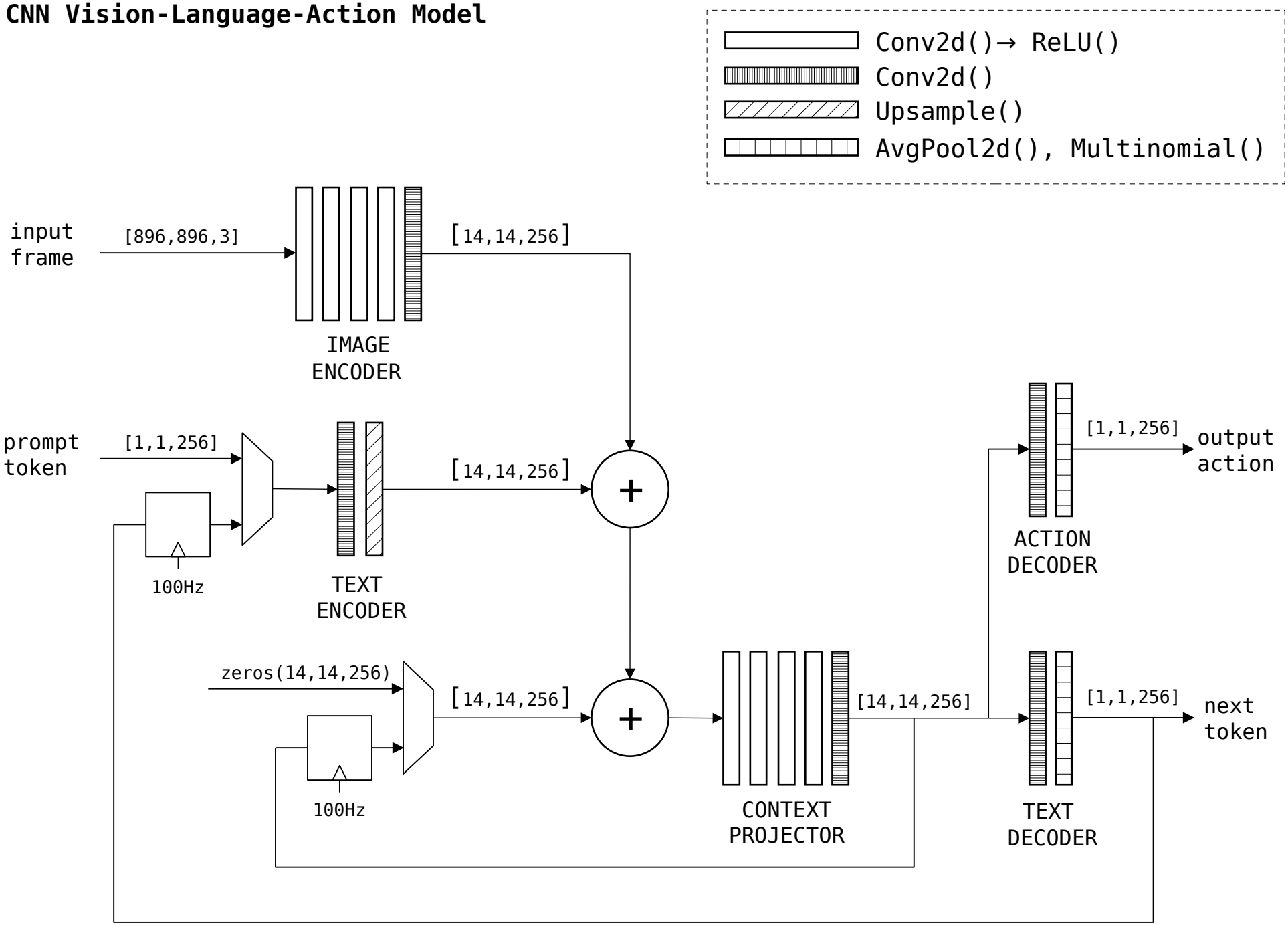
Embedded Robot Controller



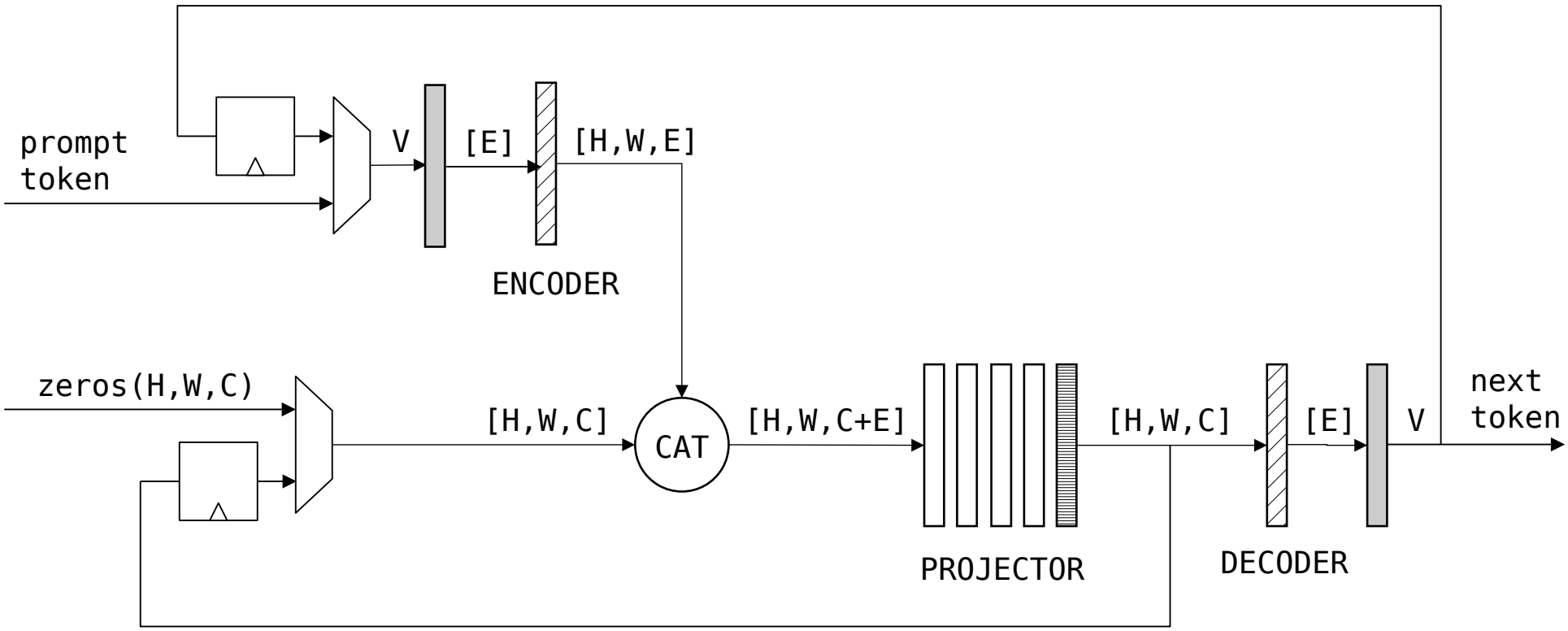
embedded components vs. AI accelerator
real time performance vs. flexibility

CNN Vision-Language-Action Model



BACKUP

CNN Language Model



Conv2d()→ ReLU()

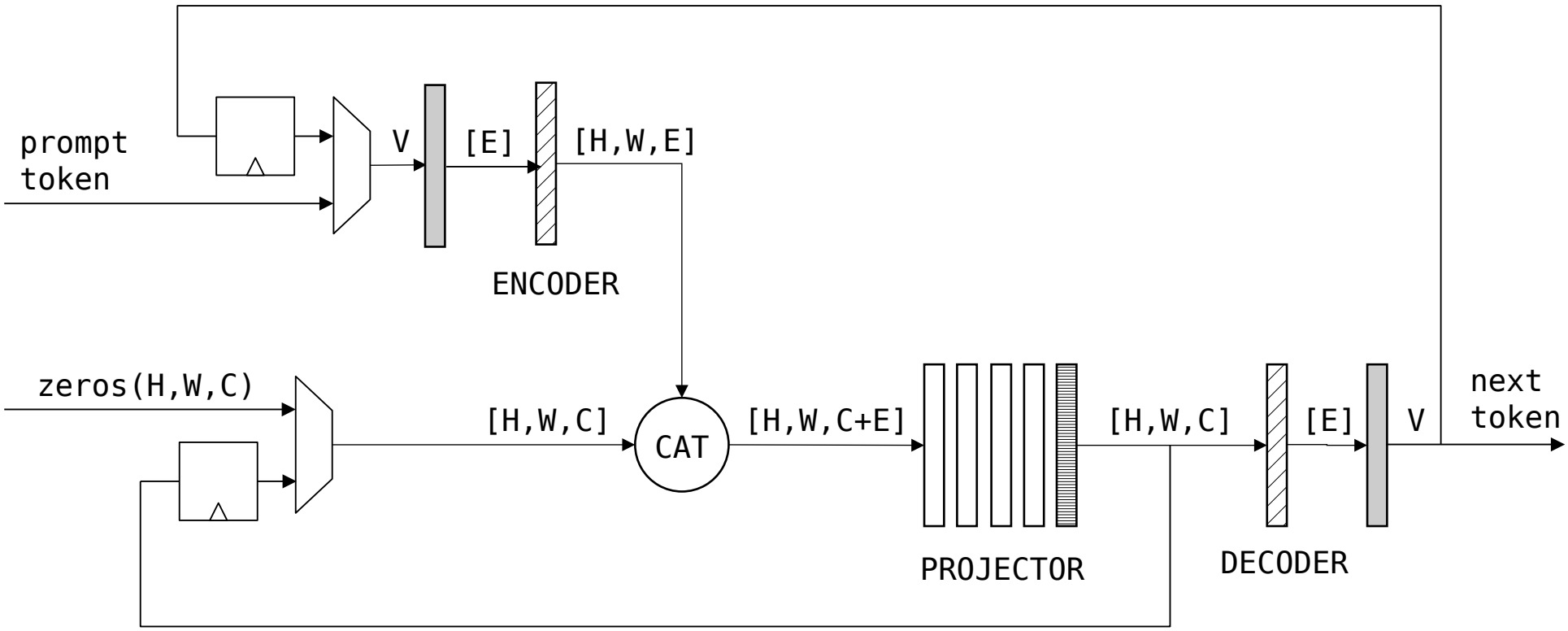
Conv2d()

Upsample(), AvgPool2d()

FROZEN

V	50257
E	256
C	512
H,W	8,8

CNN Language Model



Conv2d()→ ReLU()

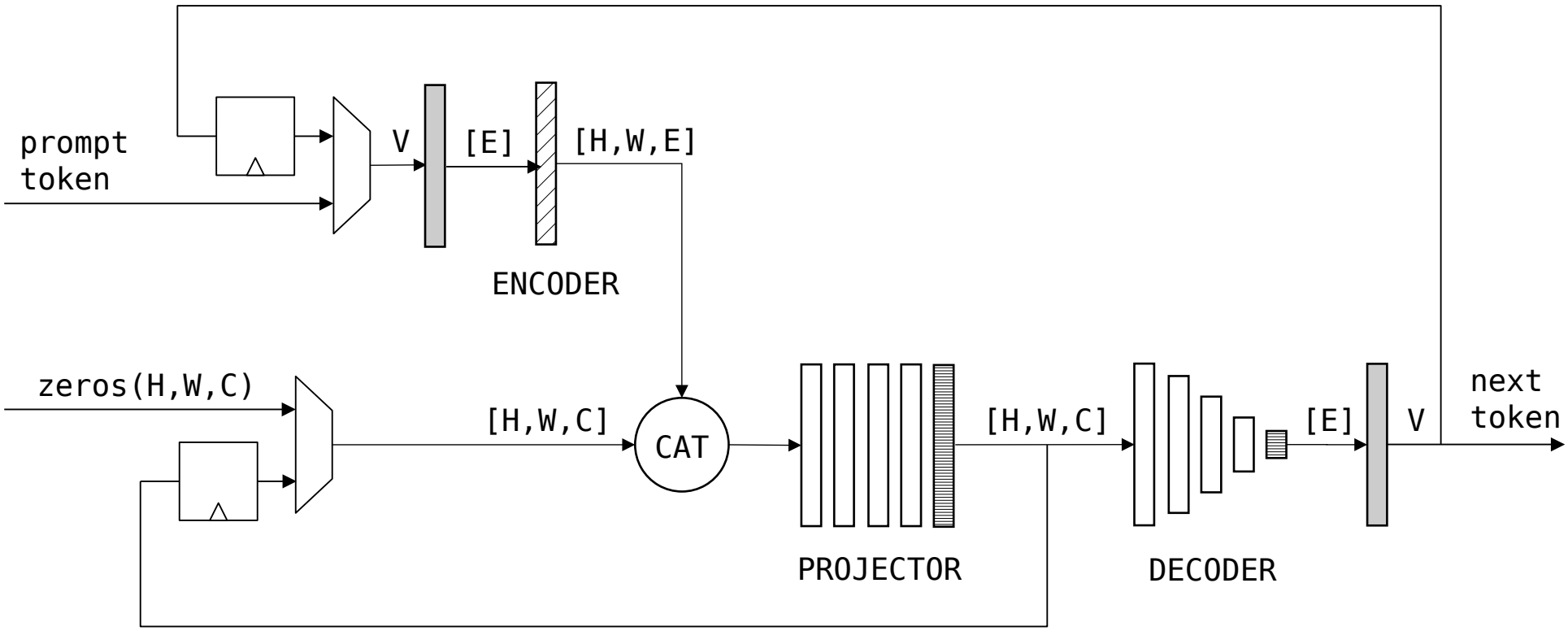
Conv2d()

Upsample(), AvgPool2d()

FROZEN

V	50257
E	256
C	512
H,W	8,8

CNN Language Model



Conv2d()→ ReLU()

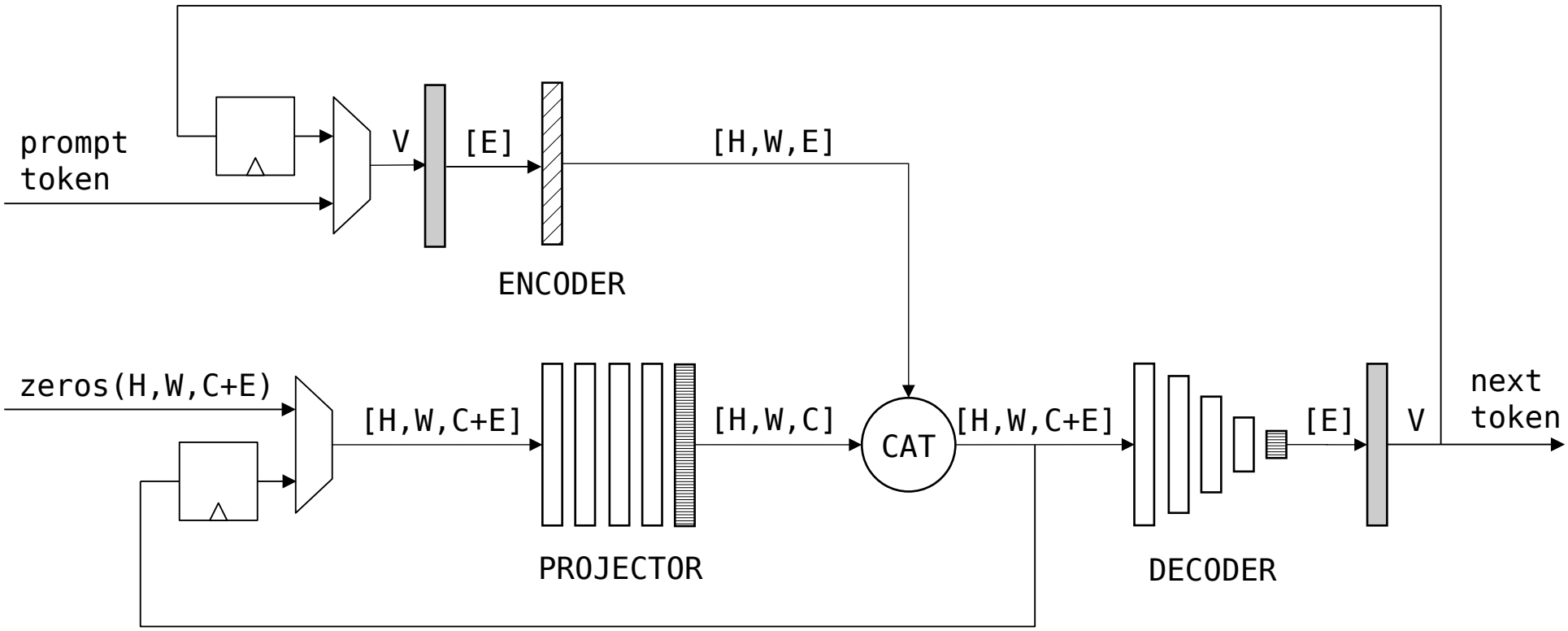
Conv2d()

Upsample()

FROZEN

E	256
V	50257
C	256
H,W	14,14

CNN Language Model



Conv2d()→ ReLU()

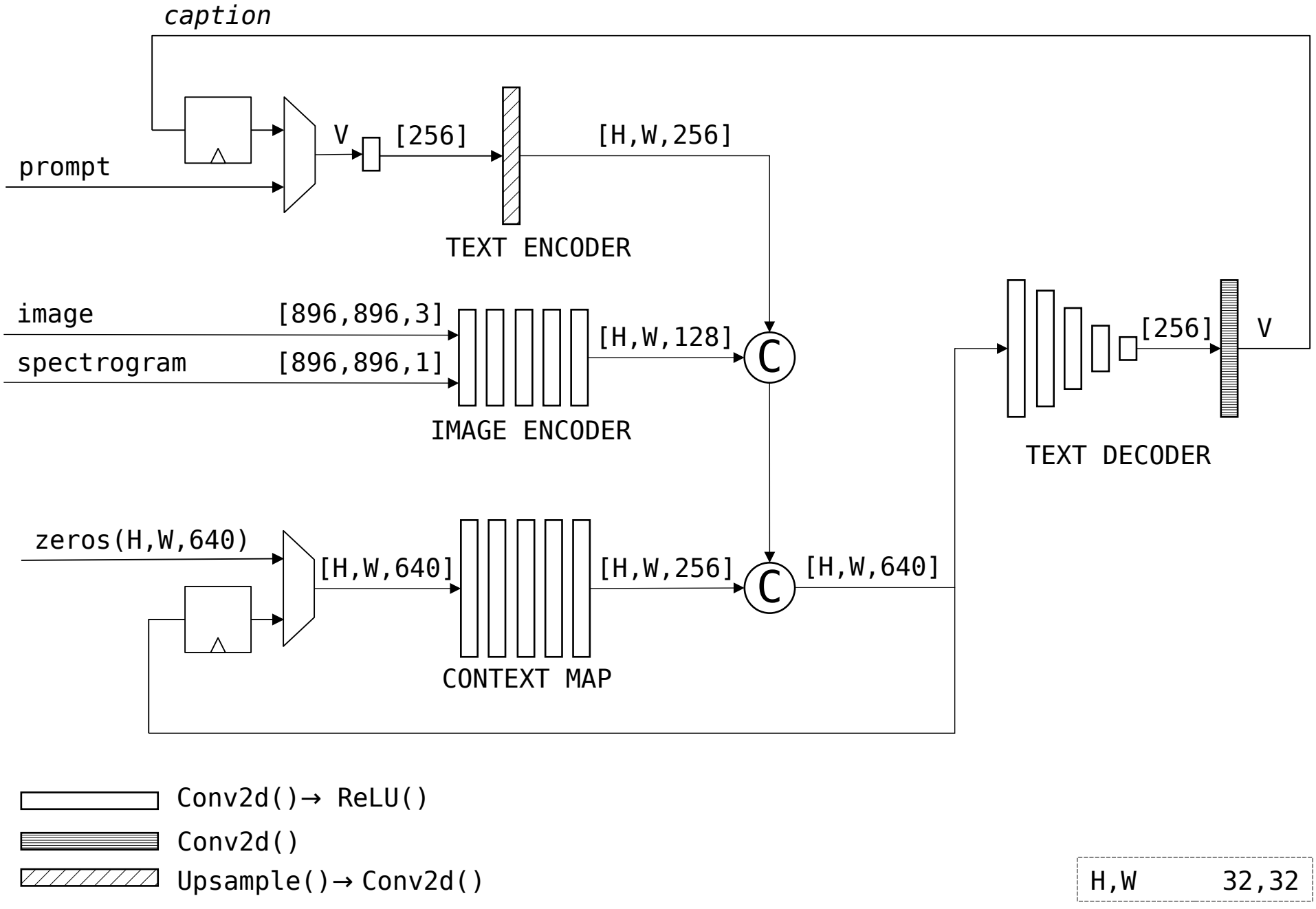
Conv2d()

Upsample()

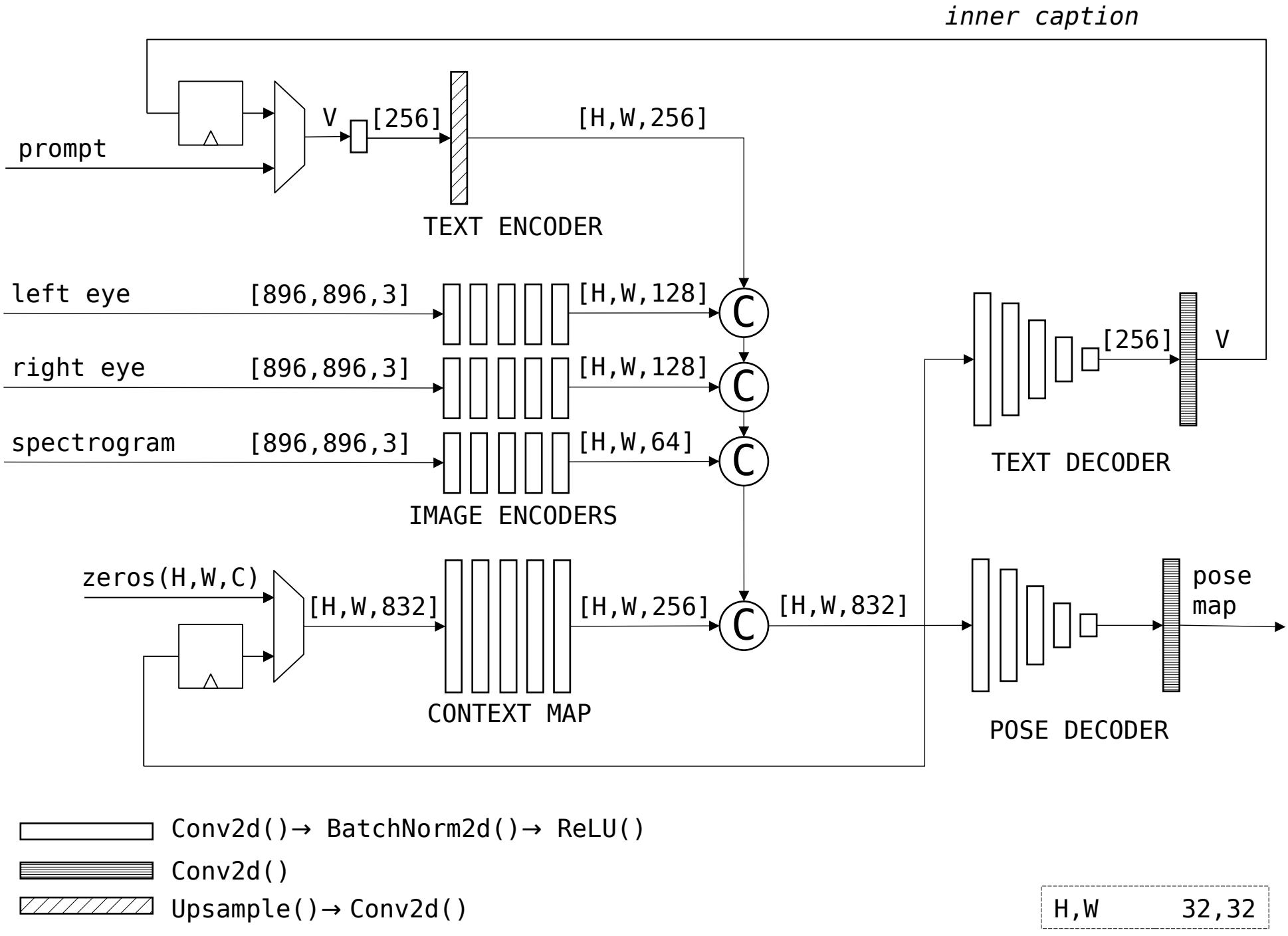
FROZEN

E	256
V	50257
C	256
H,W	14,14

CNN Vision-Language Model



CNN Perception-Language-Action Model



PHASE 1:
CNN Perception-Language-Action model (CNN-PLA), running on GPU
>20fps, <100ms latency, <400W

PHASE 2:
CNN-PLA model running on FPGA
>100fps, <15ms latency, <100W

PHASE 3:
CNN-PLA model running on ASIC at >300fps, <5ms latency, <10W

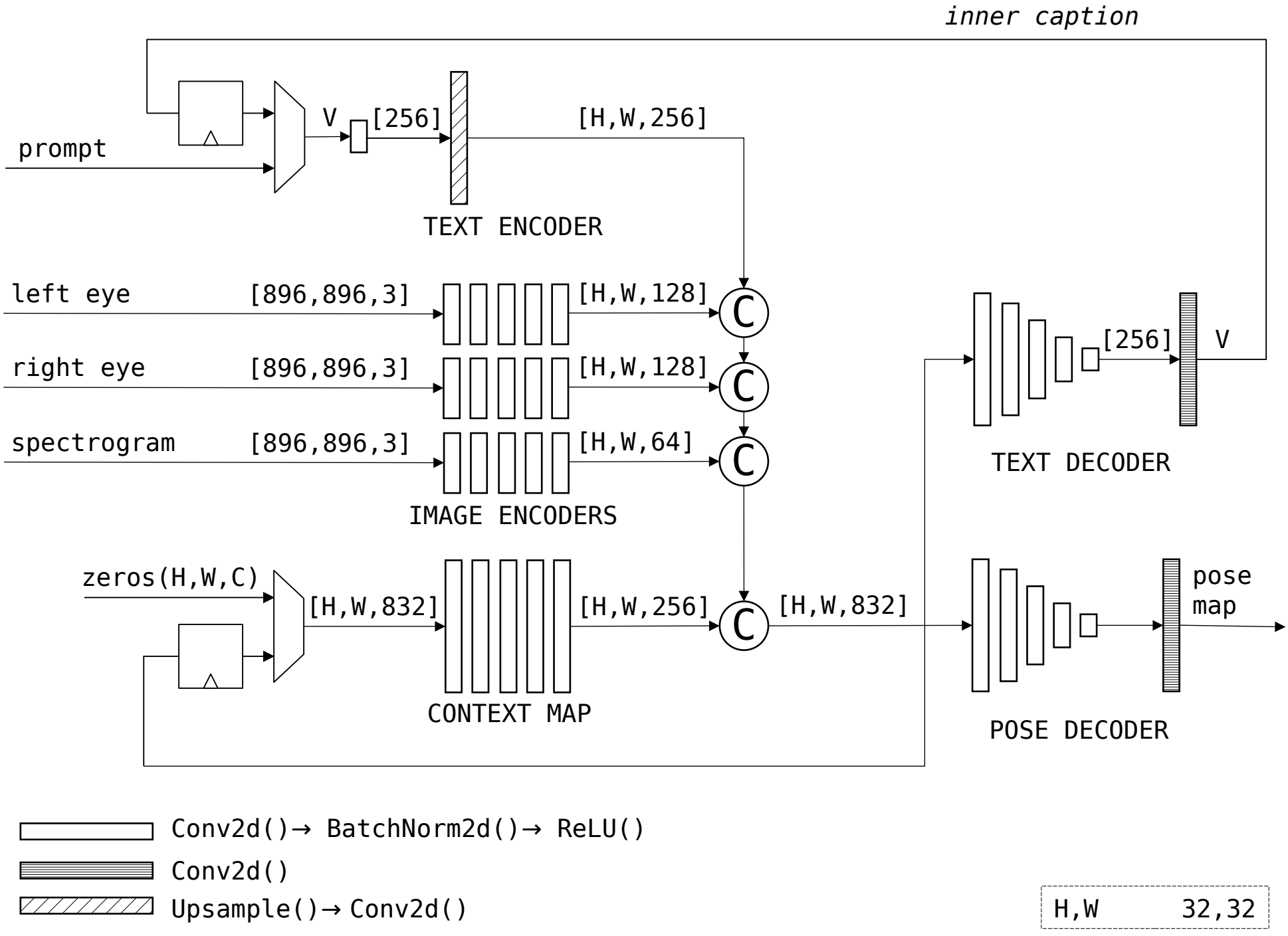
Seated humanoid :
1-DoF spine
2-DoF shoulder
2-DoF neck
6-DoF arms
6-DoF hands

Data collection :
VR teleoperated
spine/shoulder : swivel chair with up/down control
neck : VR gyroscope
arms/hands : TBD

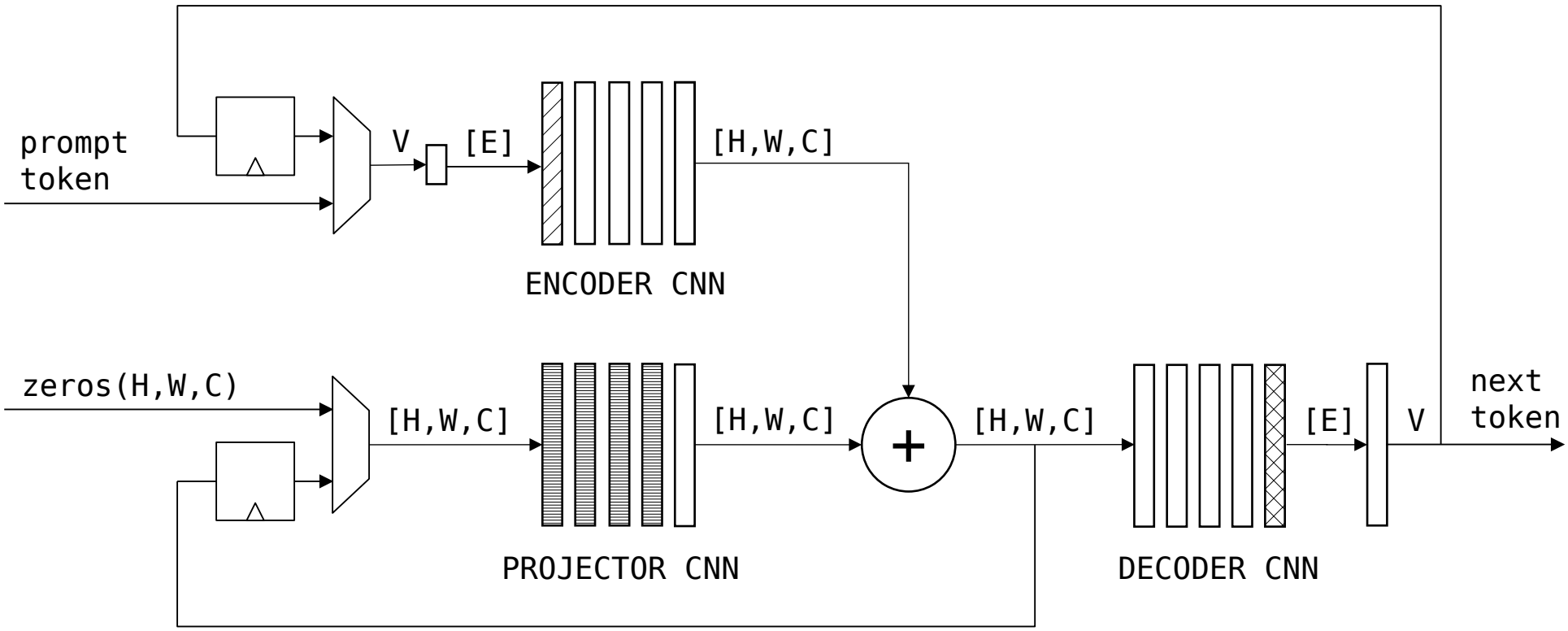
ROBOT
896x896 RGB at 120Hz, left and right eye
896x896 spectrogram at 120Hz, audio and tactile signals
29-DoF target pose at 120Hz

TELEOPERATOR
transcribed verbal commentary at 120 characters/s
synthetically captioned using open source VL model, 1 character/frame

CNN Perception-Language-Action Model



CNN Language Model



Conv2d()→ ReLU()

Conv2d()

Upsample()

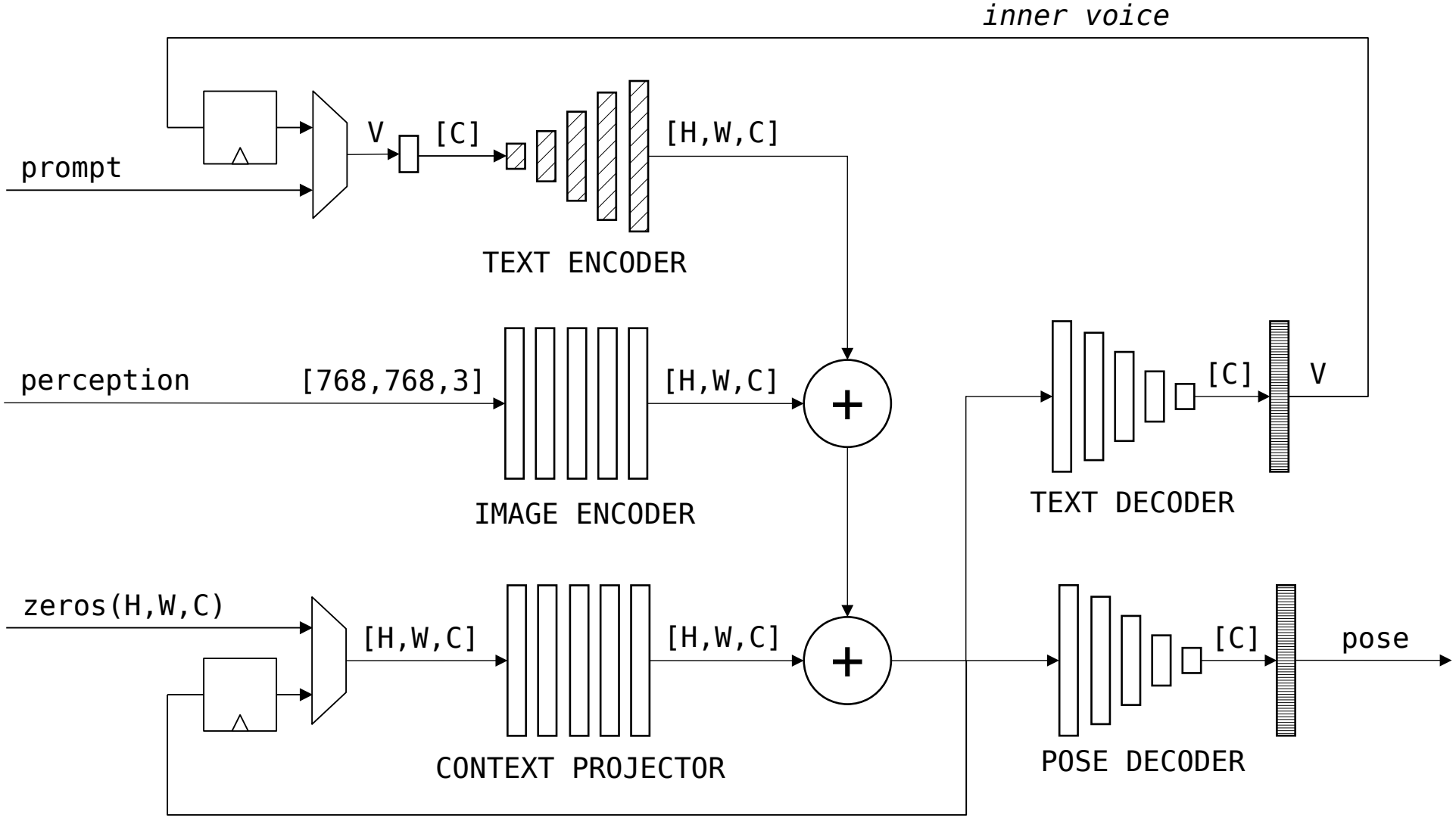
AvgPool()

E	256
V	50257
C	384
H,W	14,14

Devices

- TE256A : Token Encoder, vocab_size=256, ASCII
- TD256A : Token Decoder, vocab_size=256, ASCII
- CP384V : Context Projector, n_embd=384, VGG
- IE384R : Image Encoder, n_embd=384, Resnet
- TD
-

alt = lite-base-resnet-xeno



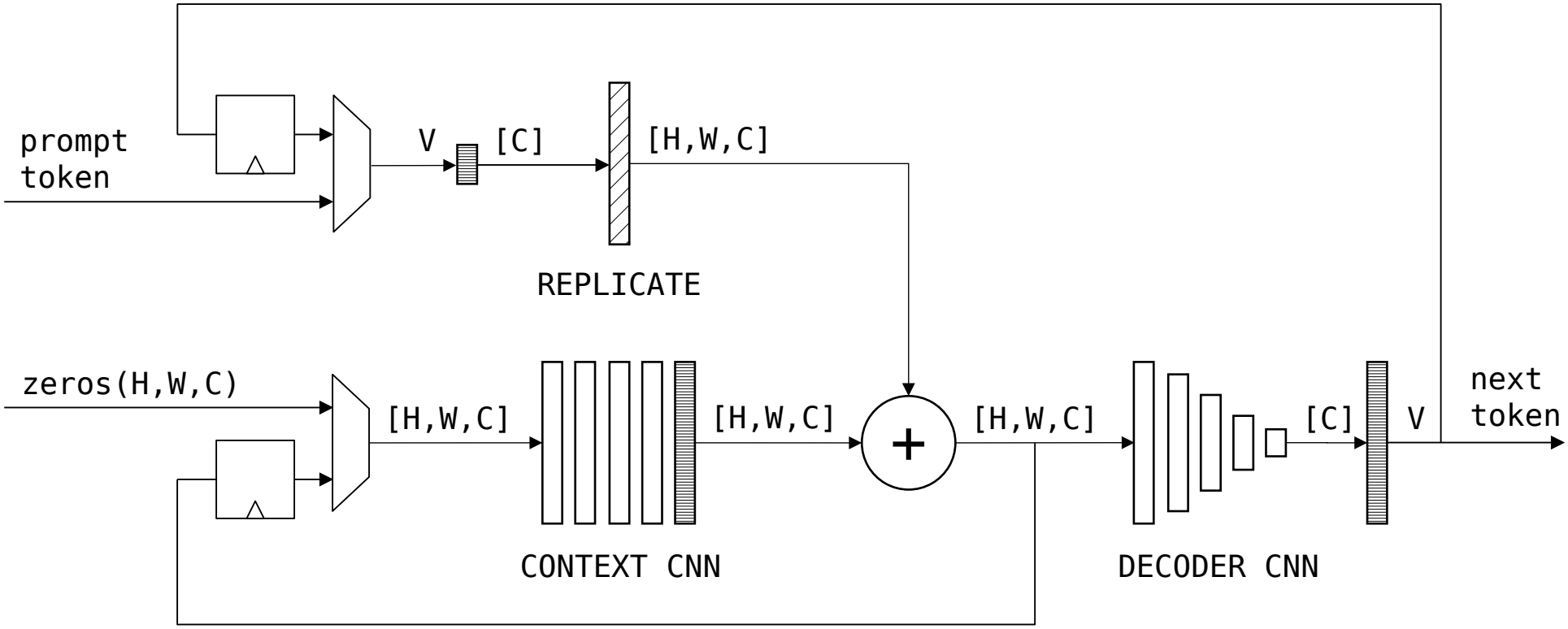
Conv2d(kernel=3)→ ReLU()

Conv2d(kernel=1)

Upsample(scale=3)→ Conv2d(kernel=3)→ ReLU()

V	256
C	384
H,W	81,81

alt2



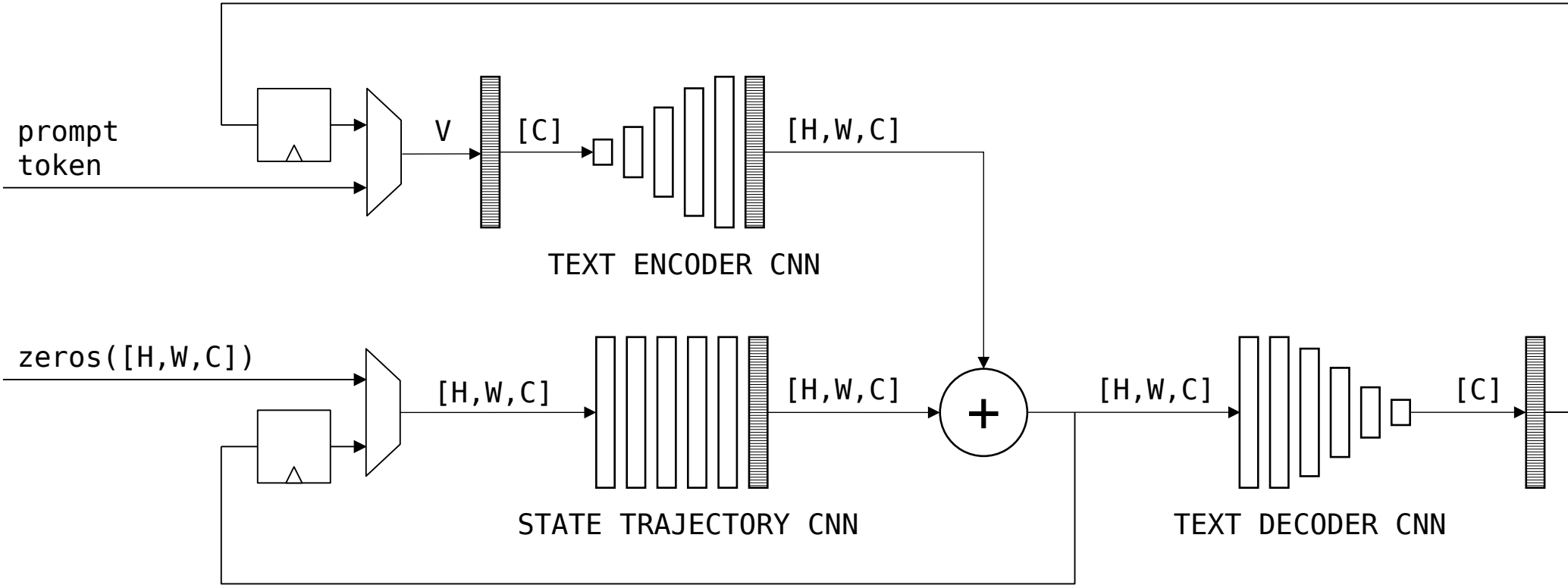
Conv2d(kernel=3)→ BatchNorm2d()→ ReLU()

Conv2d(kernel=1)

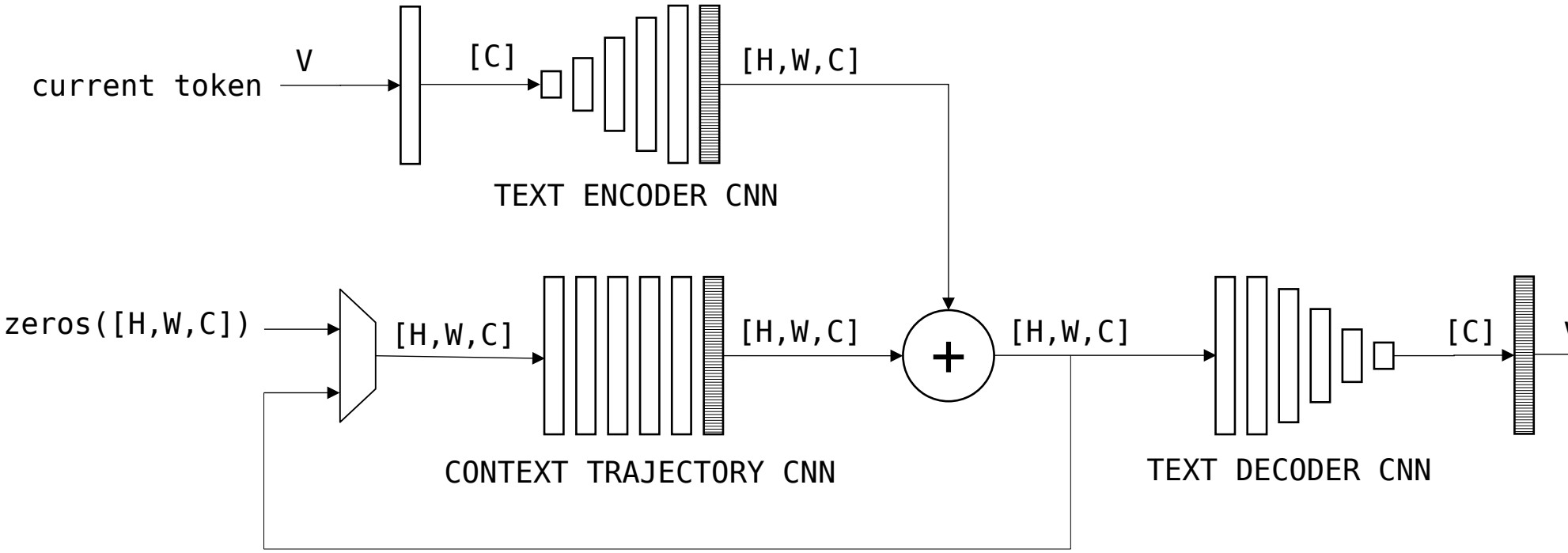
Upsample()

V	256
C	384
H,W	81,81

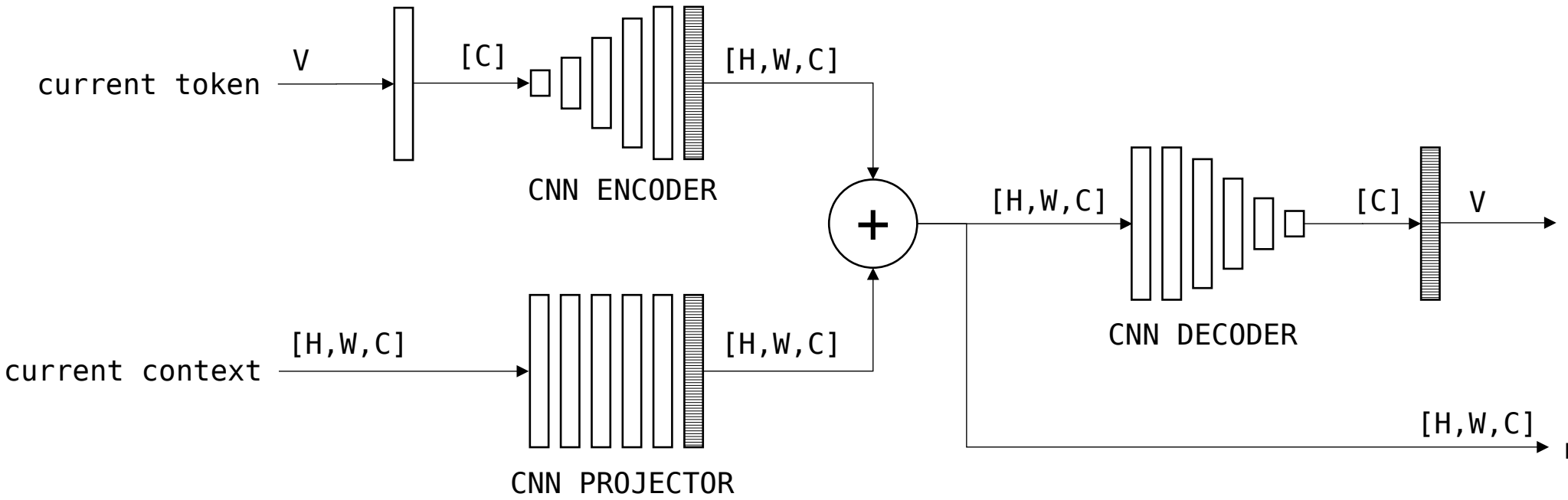
alt1


prompt
token
V
[C]
[H,W,C]
TEXT ENCODER CNN
zeros([H,W,C])
[H,W,C]
[H,W,C]
+
[H,W,C]
[C]
STATE TRAJECTORY CNN
TEXT DECODER CNN

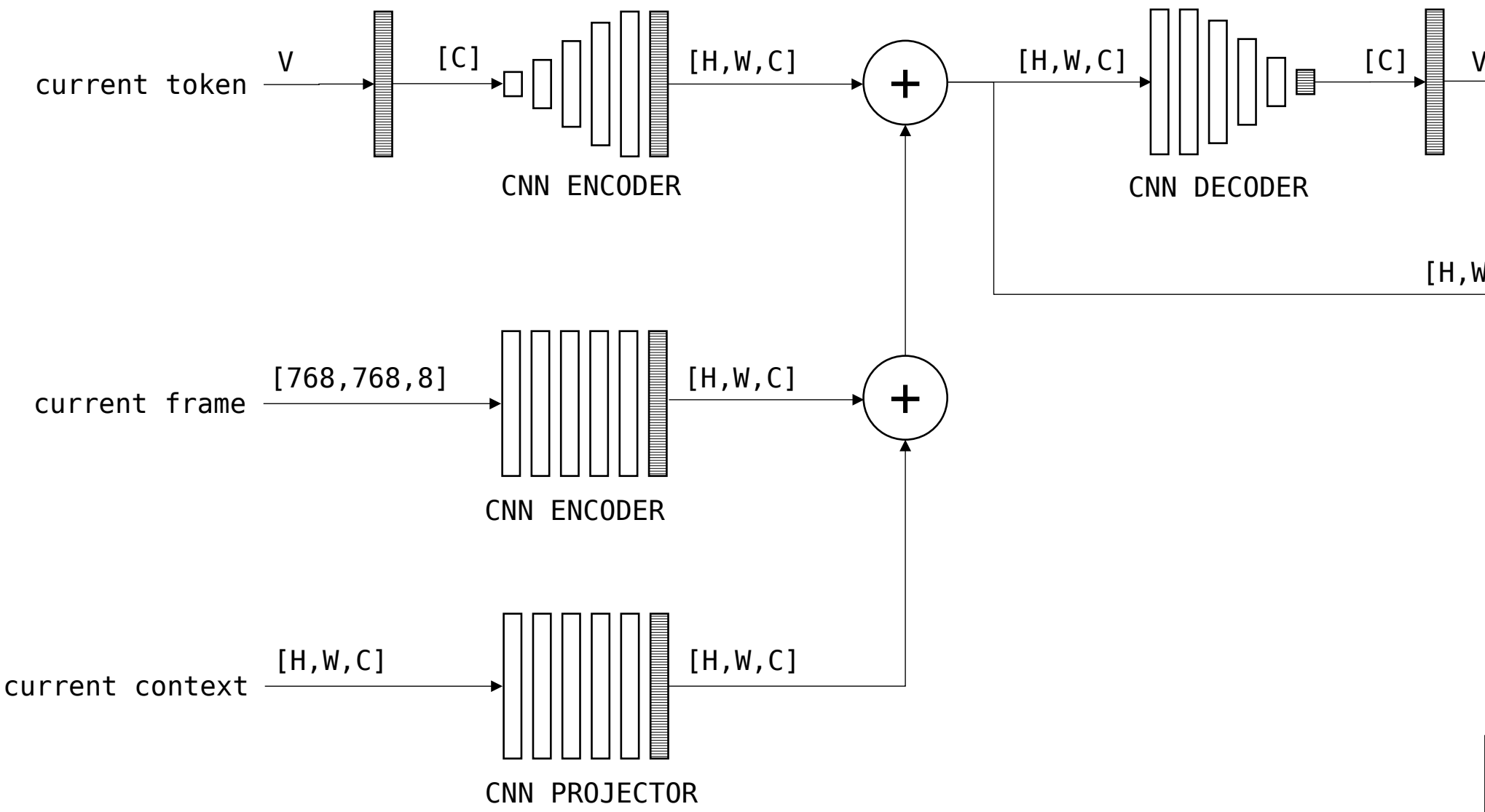
alt1


current token
V
[C]
[H,W,C]
TEXT ENCODER CNN
zeros([H,W,C])
[H,W,C]
[H,W,C]
+
[H,W,C]
[C]
CONTEXT TRAJECTORY CNN
TEXT DECODER CNN

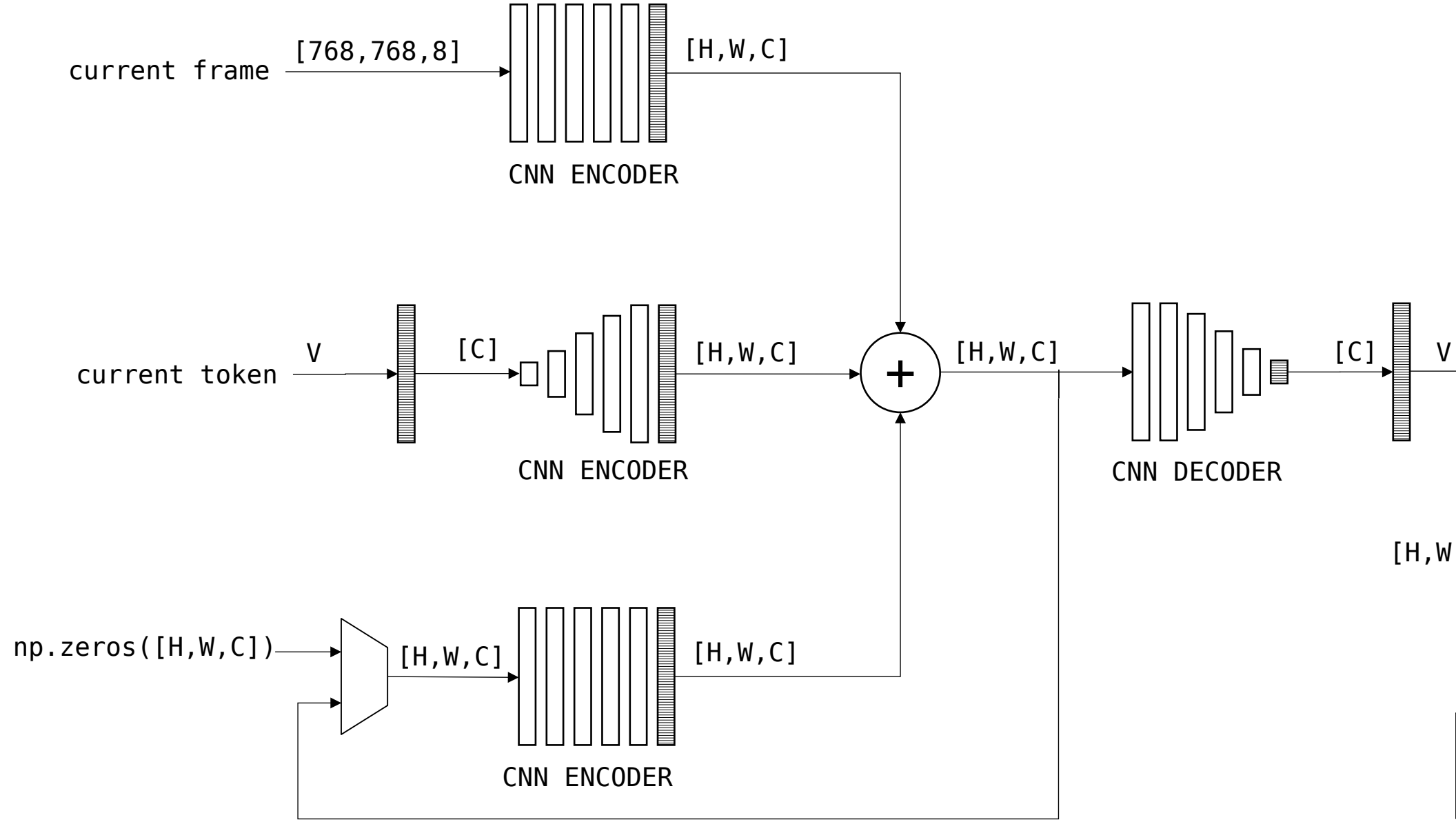
alt1


current token
V
[C]
[H,W,C]
CNN ENCODER
current context
[H,W,C]
[H,W,C]
CNN PROJECTOR
+
[H,W,C]
CNN DECODER
[C]
V
[H,W,C]

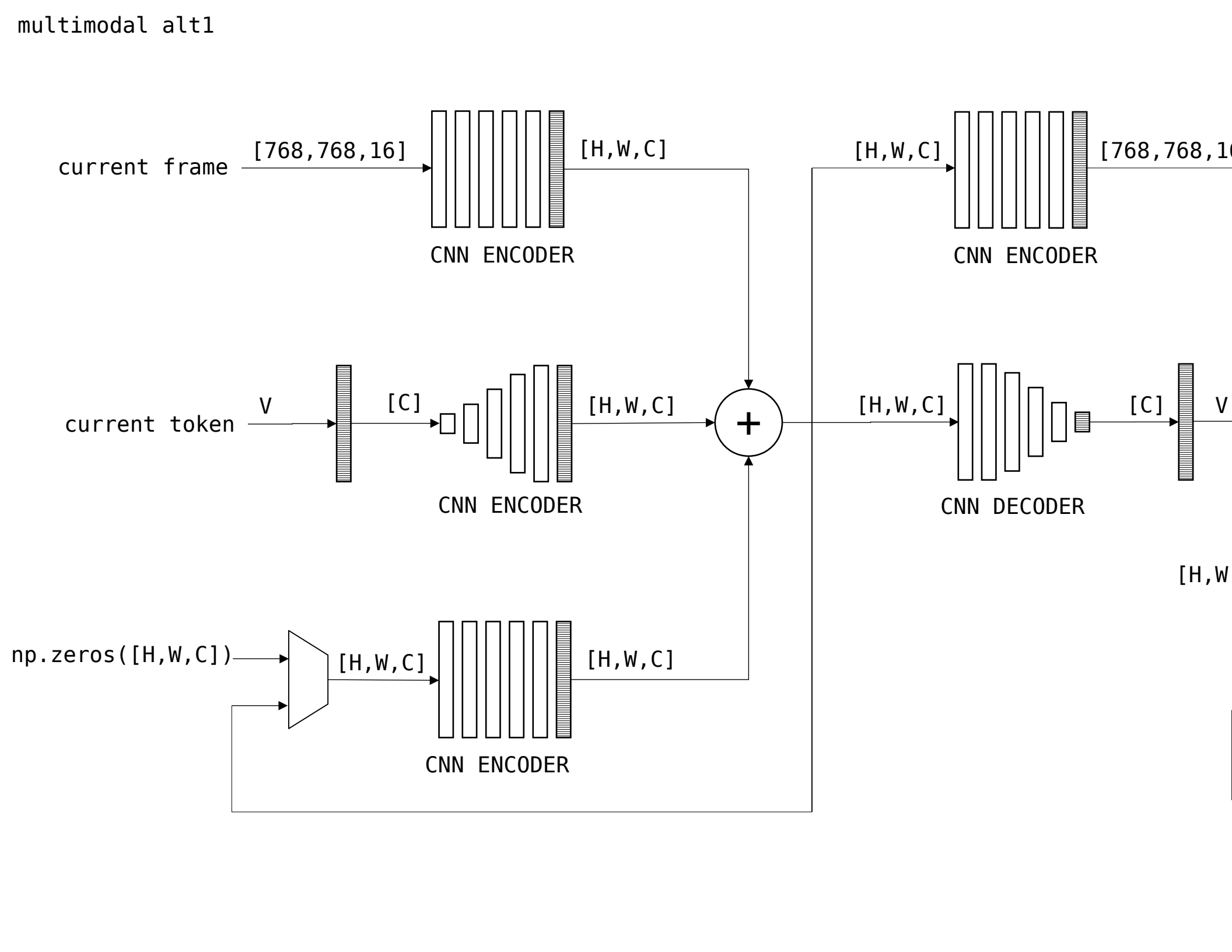
multimodal alt1



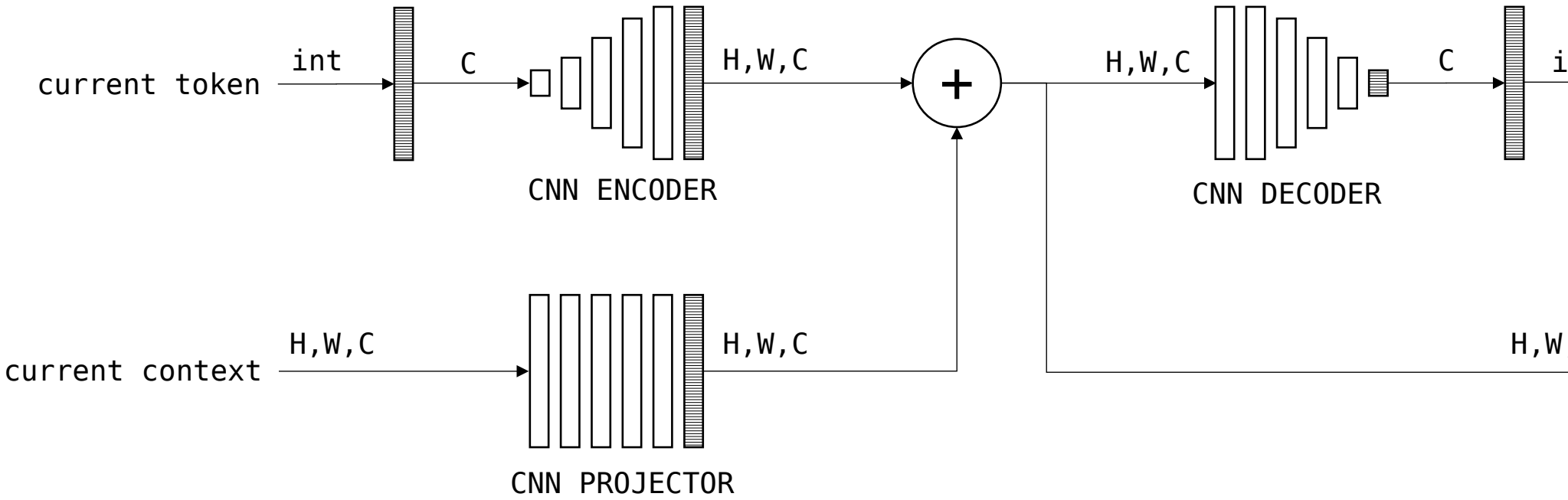
multimodal alt1

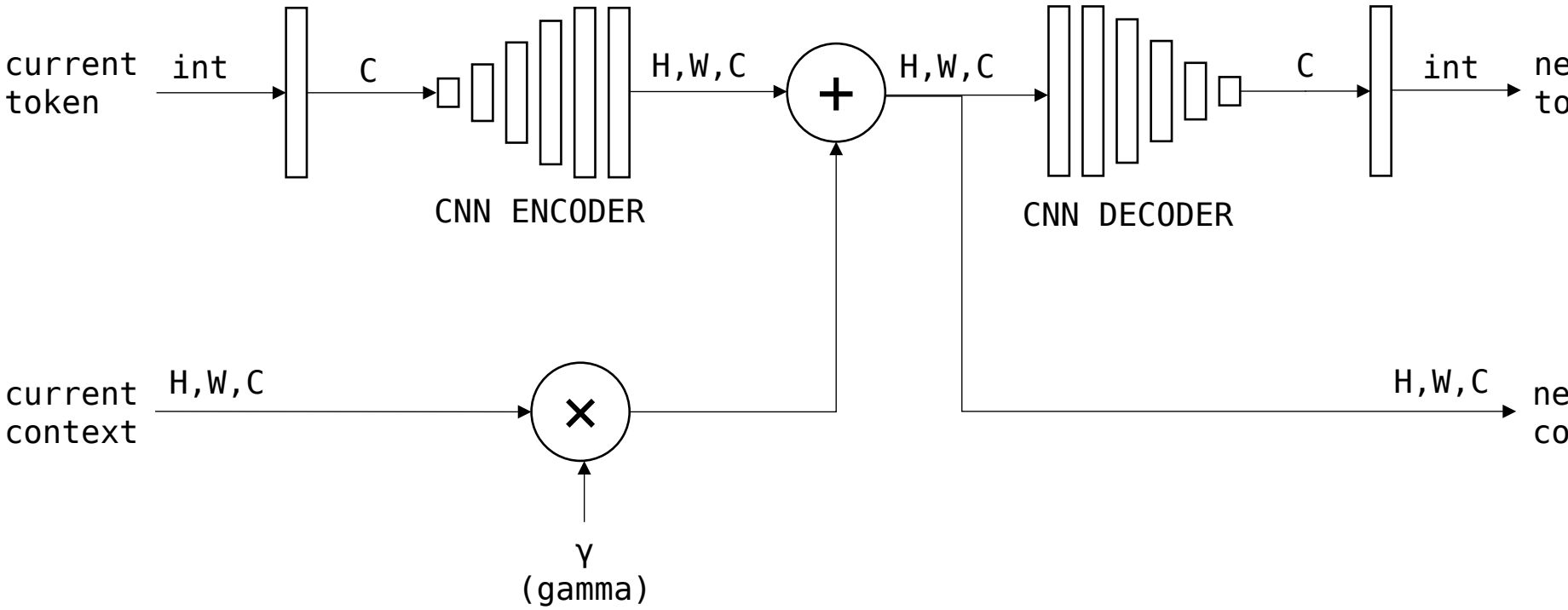


multimodal alt1

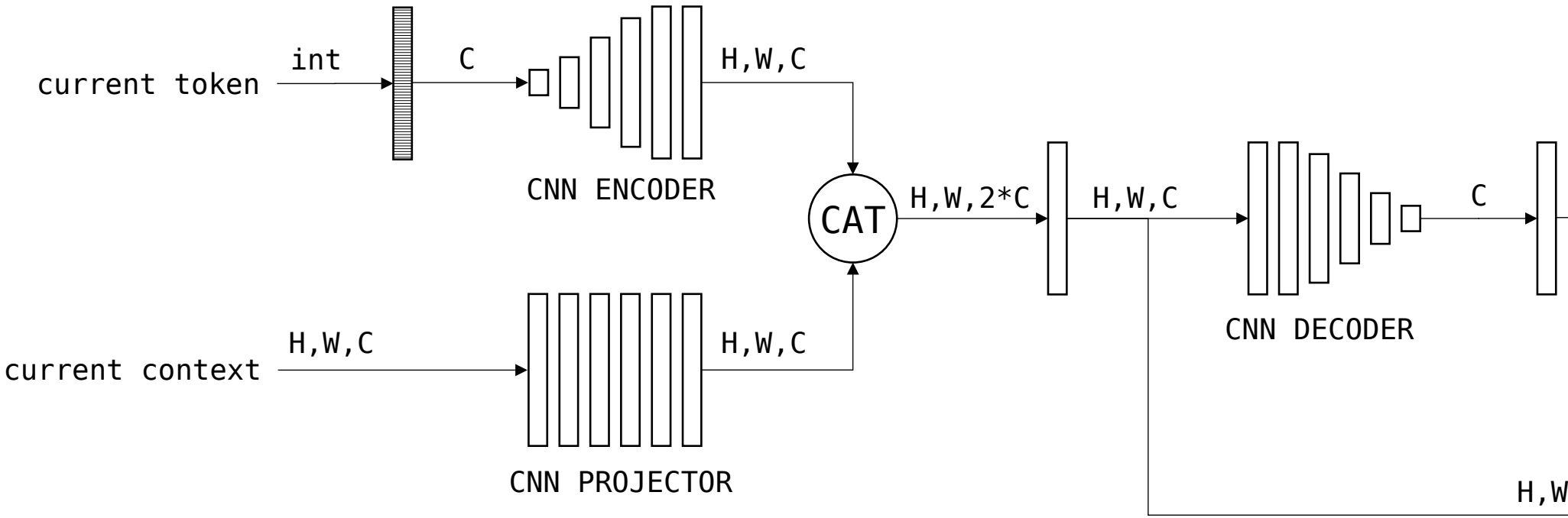


alt1

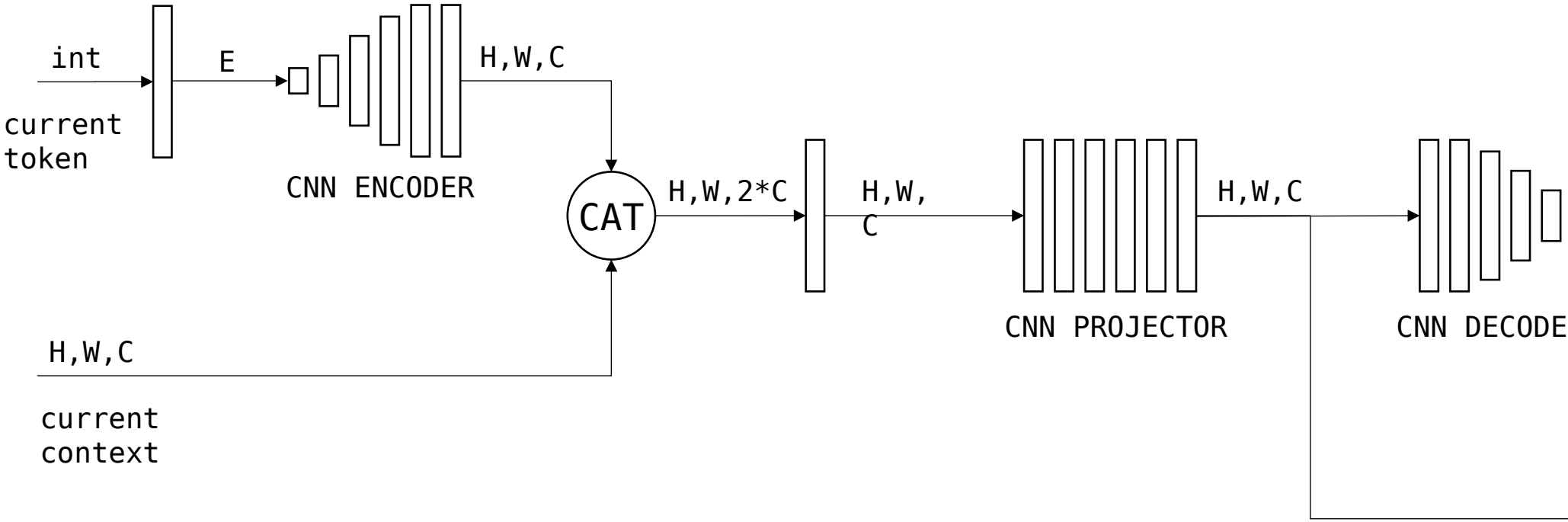

current token
int
C
H,W,C
CNN ENCODER
H,W,C
C
i
CNN DECODER
current context
H,W,C
H,W,C
H,W
CNN PROJECTOR

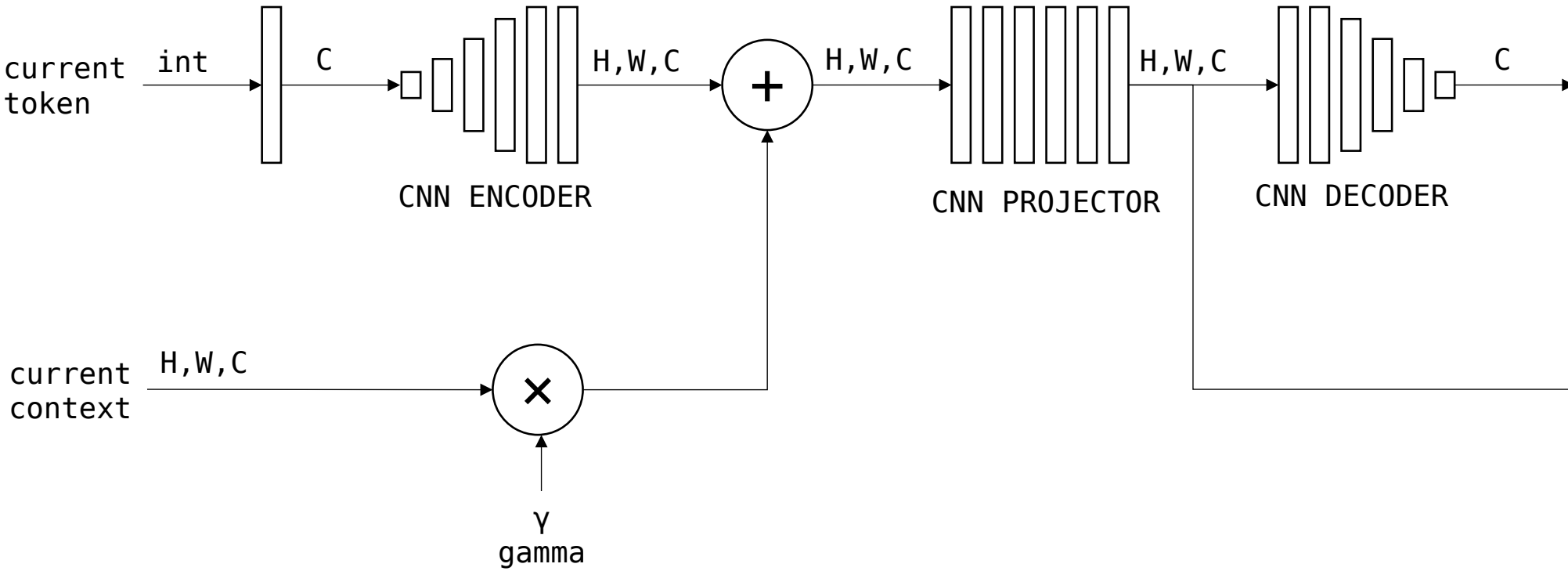

current
token
int
C
H,W,C
CNN ENCODER
H,W,C
CNN DECODER
C
int
current
context
H,W,C
γ
(gamma)
H,W,C

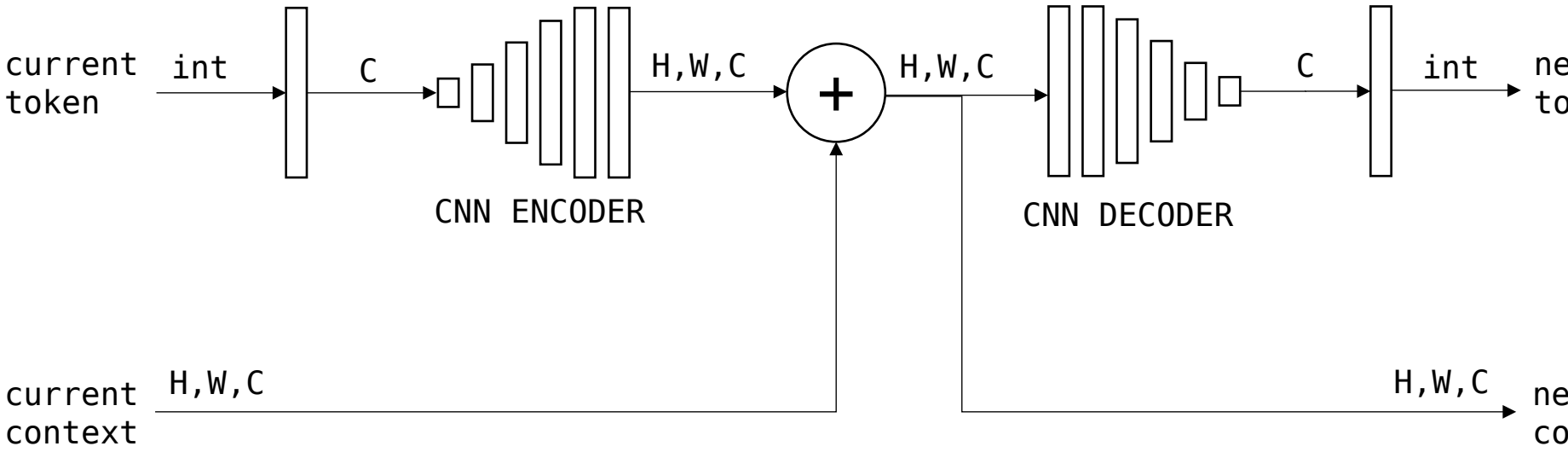
alt2

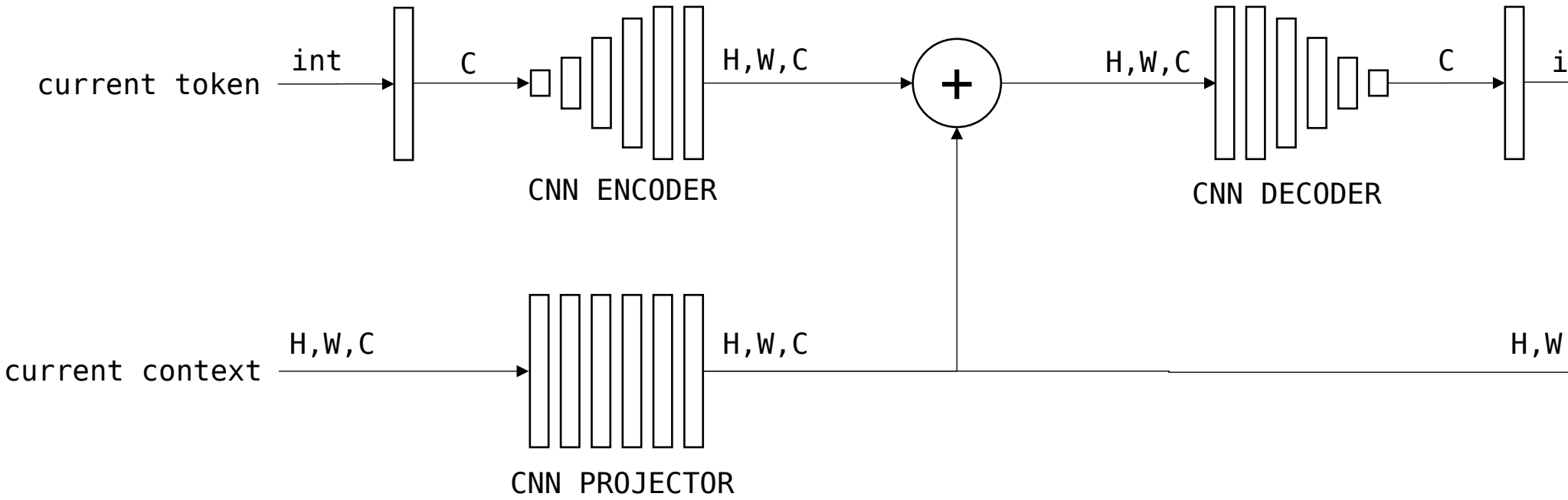

current token
int
C
H,W,C
CNN ENCODER
CAT
H,W,2*C
H,W,C
CNN DECODER
C
current context
H,W,C
H,W,C
CNN PROJECTOR
H,W

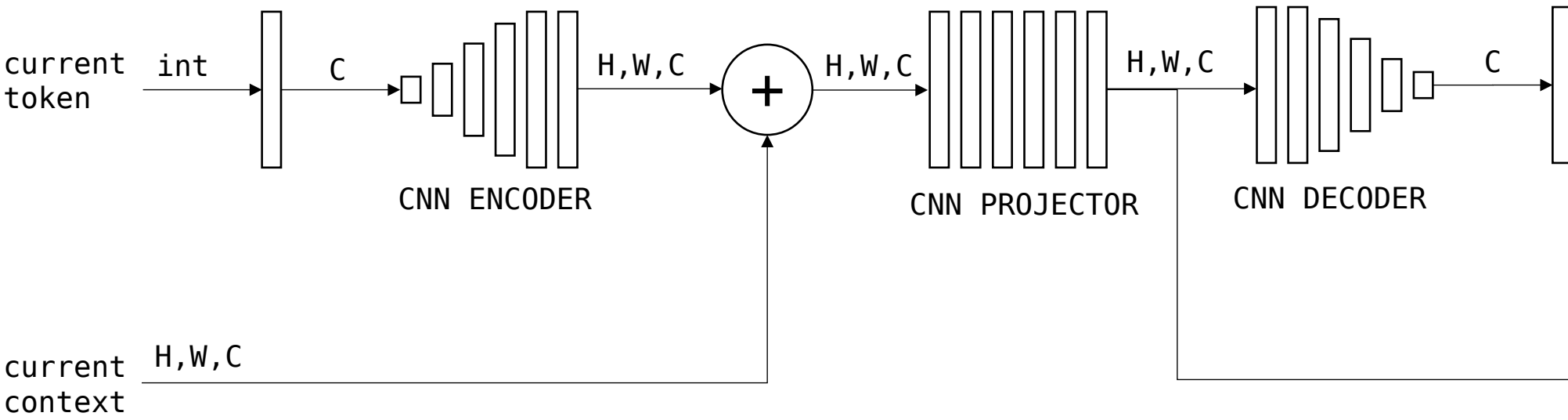
alt3

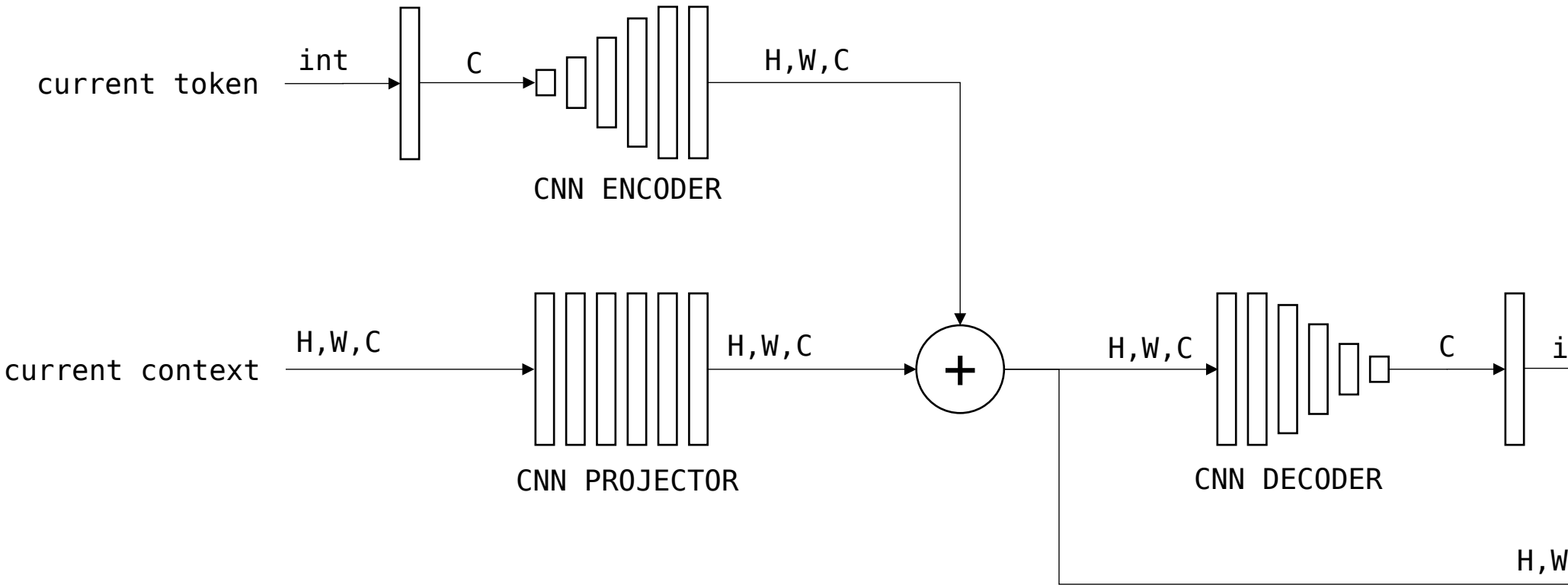

int
current
token
E
H,W,C
CNN ENCODER
CAT
H,W,2*C
H,W,
C
CNN PROJECTOR
H,W,C
CNN DECODE
H,W,C
current
context


current
token
int
C
H,W,C
CNN ENCODER
H,W,C
H,W,C
CNN PROJECTOR
H,W,C
CNN DECODER
C
current
context
H,W,C
γ
gamma


current
token
int
C
H,W,C
H,W,C
C
int
ne
to
CNN ENCODER
CNN DECODER
current
context
H,W,C
H,W,C
ne
co


current token
int
C
H,W,C
CNN ENCODER
H,W,C
CNN DECODER
C
i
current context
H,W,C
H,W,C
CNN PROJECTOR
H,W


current
token
int
C
H,W,C
H,W,C
H,W,C
C
CNN ENCODER
CNN PROJECTOR
CNN DECODER
current
context
H,W,C


current token
int
C
H,W,C
CNN ENCODER
current context
H,W,C
H,W,C
CNN PROJECTOR
H,W,C
C
CNN DECODER
H,W