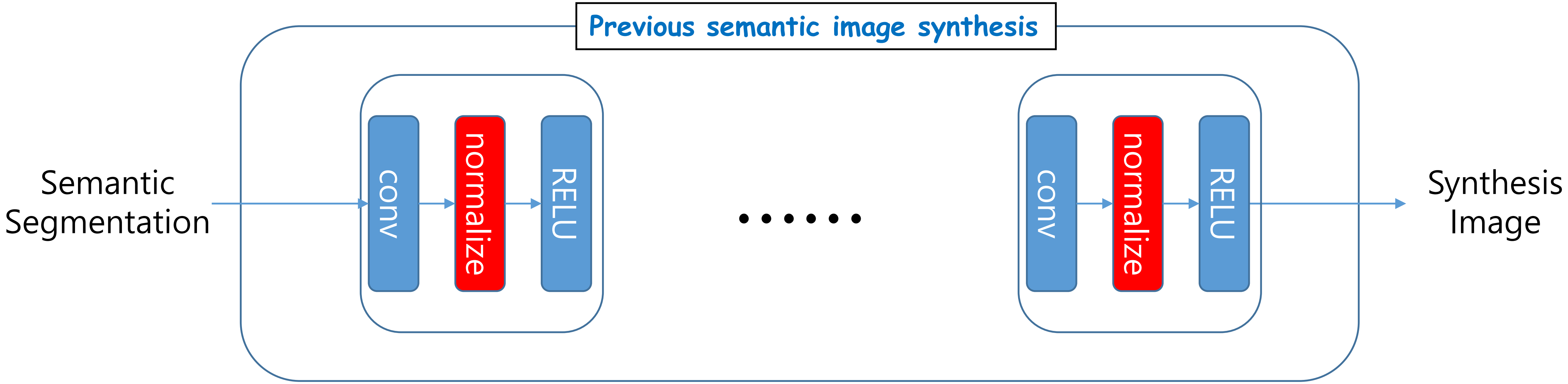
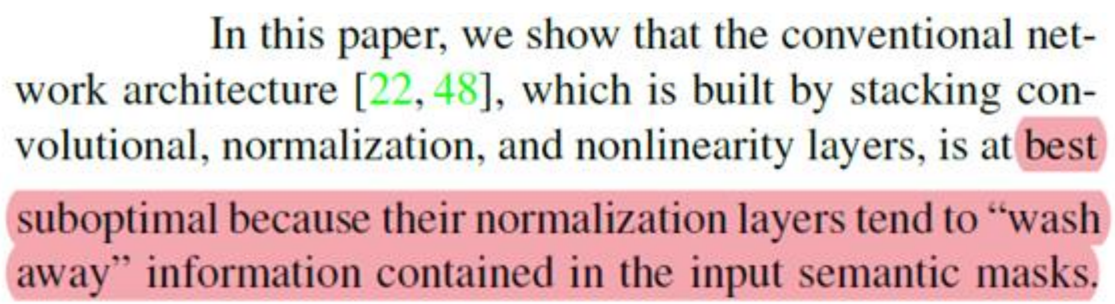
ALIAS Generator

Related work: SPADE

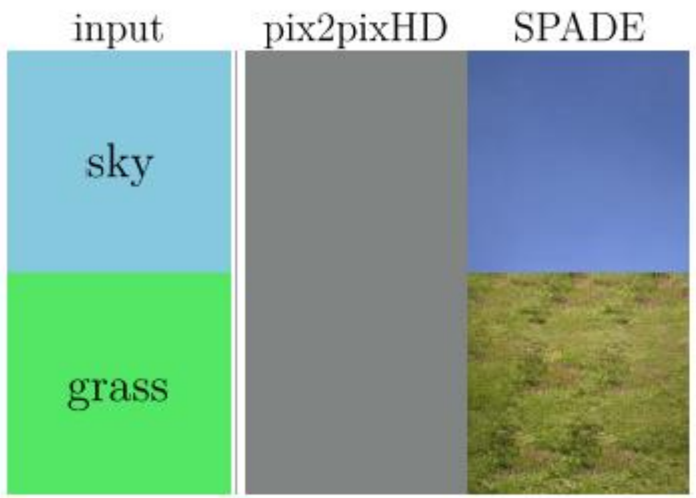



In this paper, we show that the conventional network architecture [22, 48], which is built by stacking convolutional, normalization, and nonlinearity layers, is at best suboptimal because their normalization layers tend to "wash away" information contained in the input semantic masks.

Related work: SPADE

Why does the SPADE work better? A short answer is that it can better preserve semantic information against common normalization layers. Specifically, while normalization layers such as the InstanceNorm [46] are essential pieces in almost all the state-of-the-art conditional image synthesis models [48], they tend to wash away semantic information when applied to uniform or flat segmentation masks.

Let us consider a simple module that first applies convolution to a segmentation mask and then normalization. Furthermore, let us assume that a segmentation mask with a single label is given as input to the module (e.g., all the pixels have the same label such as sky or grass). Under this setting, the convolution outputs are again uniform, with different labels having different uniform values. Now, after we apply InstanceNorm to the output, the normalized activation will become all zeros no matter what the input semantic label is given. Therefore, semantic information is totally lost.



단순 single label map을 conv + norm layer에 넣으면 0값이 나온다.

Semantic segmentation은 uniform value 들을 가지고 있으므로 정보가 손실된다

=> Semantic은 정규화 하지 말아야.

Related work: SPADE

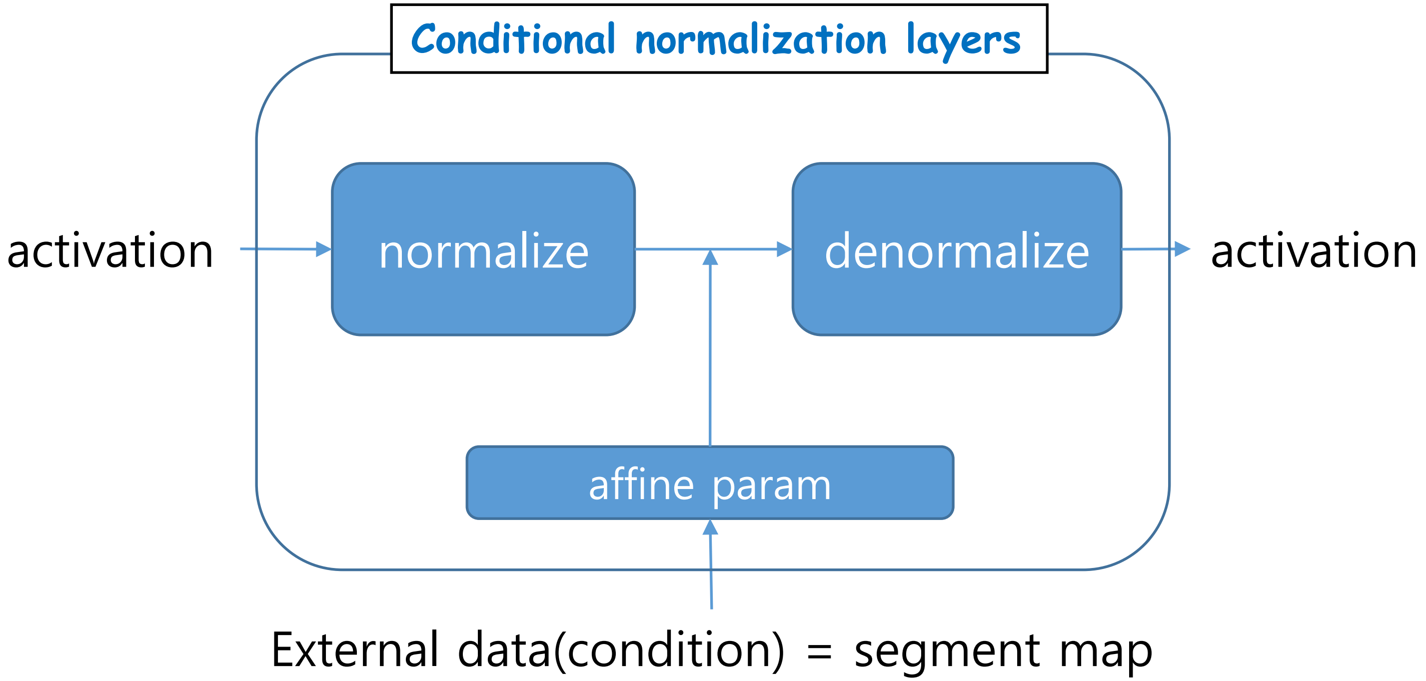
Unconditional normalization layers: activation

- Local Response Normalization
- Batch normalization
- Instance normalization
- Layer normalization
- Group normalization
- Weight normalization

Conditional normalization layers: activation + external data

- Conditional batch normalization
- Adaptive instance normalization

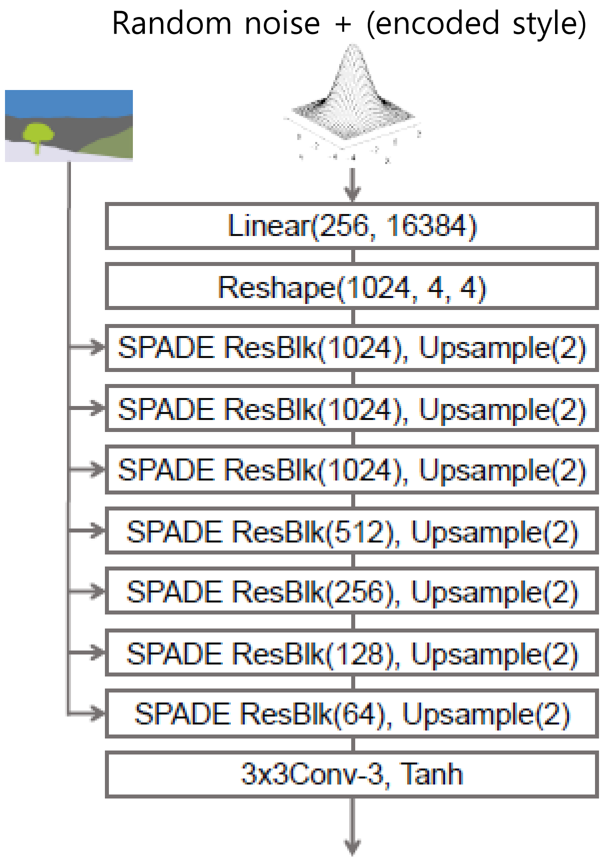
Related work: SPADE

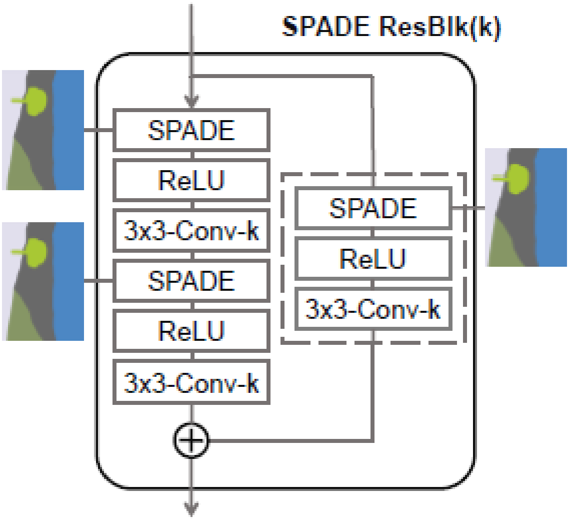


style transform(instance normalization)에서는 이미지 전체의 style을 변경하는 것이기 때문에 affine parameter가 channel별로 일정함(affine parameter가 크기가 C(channel 수)인 vector).

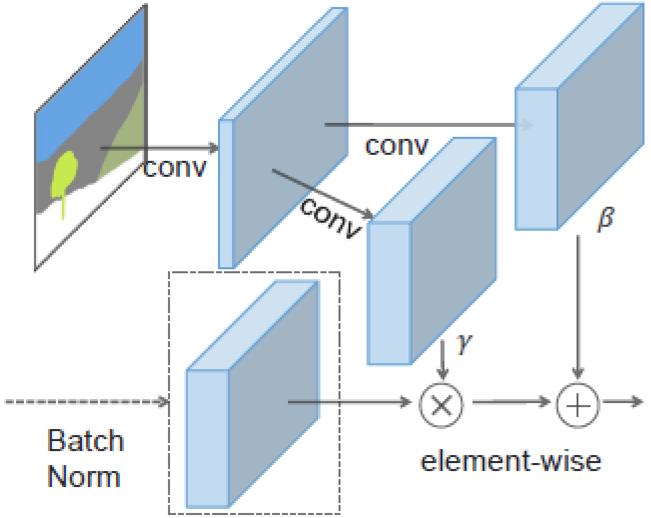
SPADE에서는 이미지의 디테일을 위해 affine parameter로 이미지를 조정한다.(affine parameter가 tensor) (spatially-varying affine parameter)

Related work: SPADE

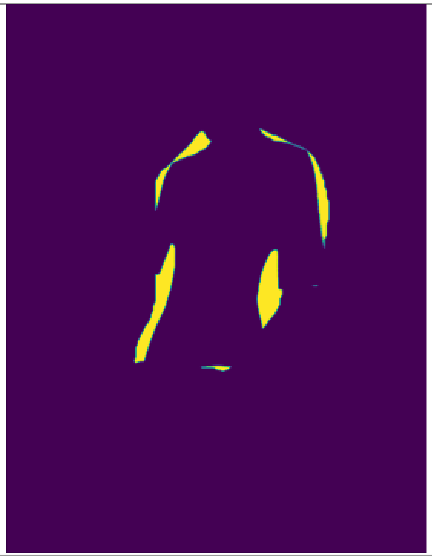
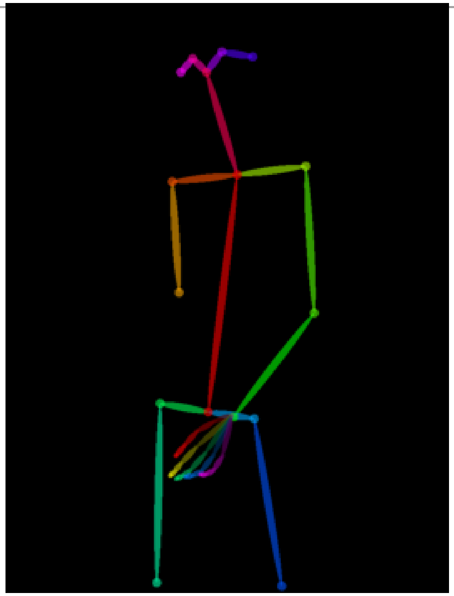




SPatially **A**daptive (**DE**)normalization



ALIAS Input

		
img_agnostic-I_a	misalign_mask-$M_{misalign}$	parse(segment)- $\hat{S}$
		
parse_div-$\hat{S}_{div}$	pose(openpose)- P	Warped_cloth-W_{cloth}

ALIAS Generator Loss

$$L_{total} = L_{Cgan} + \lambda_{FM} * L_{FM} + \lambda_{percept} * L_{percept}$$

$\lambda_{FM}, \lambda_{percept}$=10

$$L_{Cgan} = \mathbb{E}_I\left[\log\left(D(S_{div}, I)\right)\right] + \mathbb{E}_{I,c}[1 - \log(D(S_{div}, \hat{I}))]$$

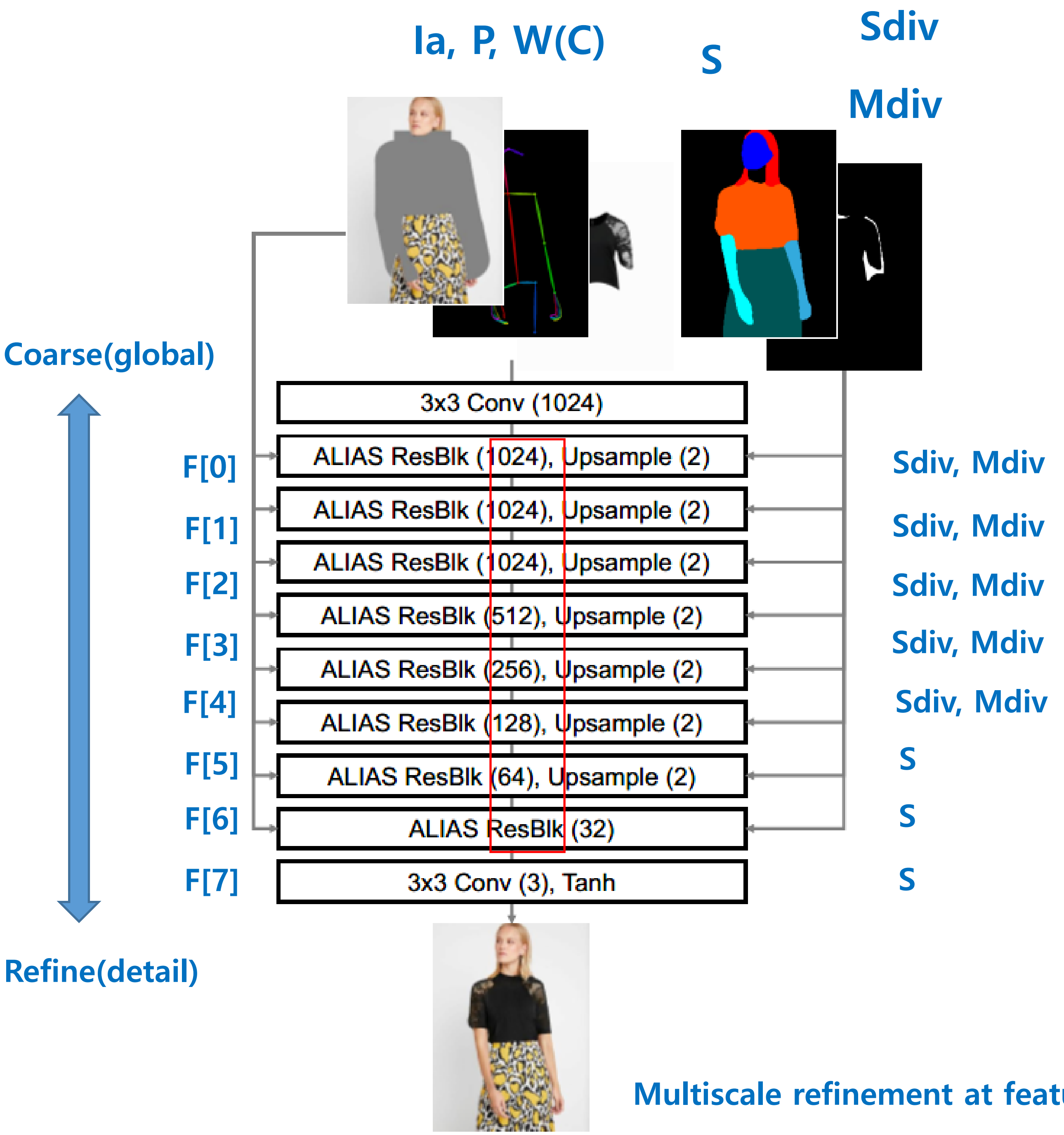
D: multi scale discriminator

$$L_{FM} = \sum_{i=1}^{T} \frac{1}{K_i} * \mathbb{E}_{I,c}[|D_I^{(i)}(S_{div}, I) - D_I^{(i)}(S_{div}, \hat{I})|]$$

$$L_{percept} = \sum_{i=1}^{T} \frac{1}{R_i} \mathbb{E}_{I,c}[|F^{(i)}(I) - F^{(i)}(\hat{I})|]$$

I: given reference(GT)
c: target cloth image
F: VGG network layer

ALIAS Generator



Default:most

conv0(1,6,8,1024)

head0(1,6,8,1024)

middle0(1,12,16,1024)

middle1(1,24,32,1024)

If more or most: add upsample layer

up0(1,48,64,512)

up1(1,96,128,256)

up2(1,192,256,128)

up3(1,384,512,64)

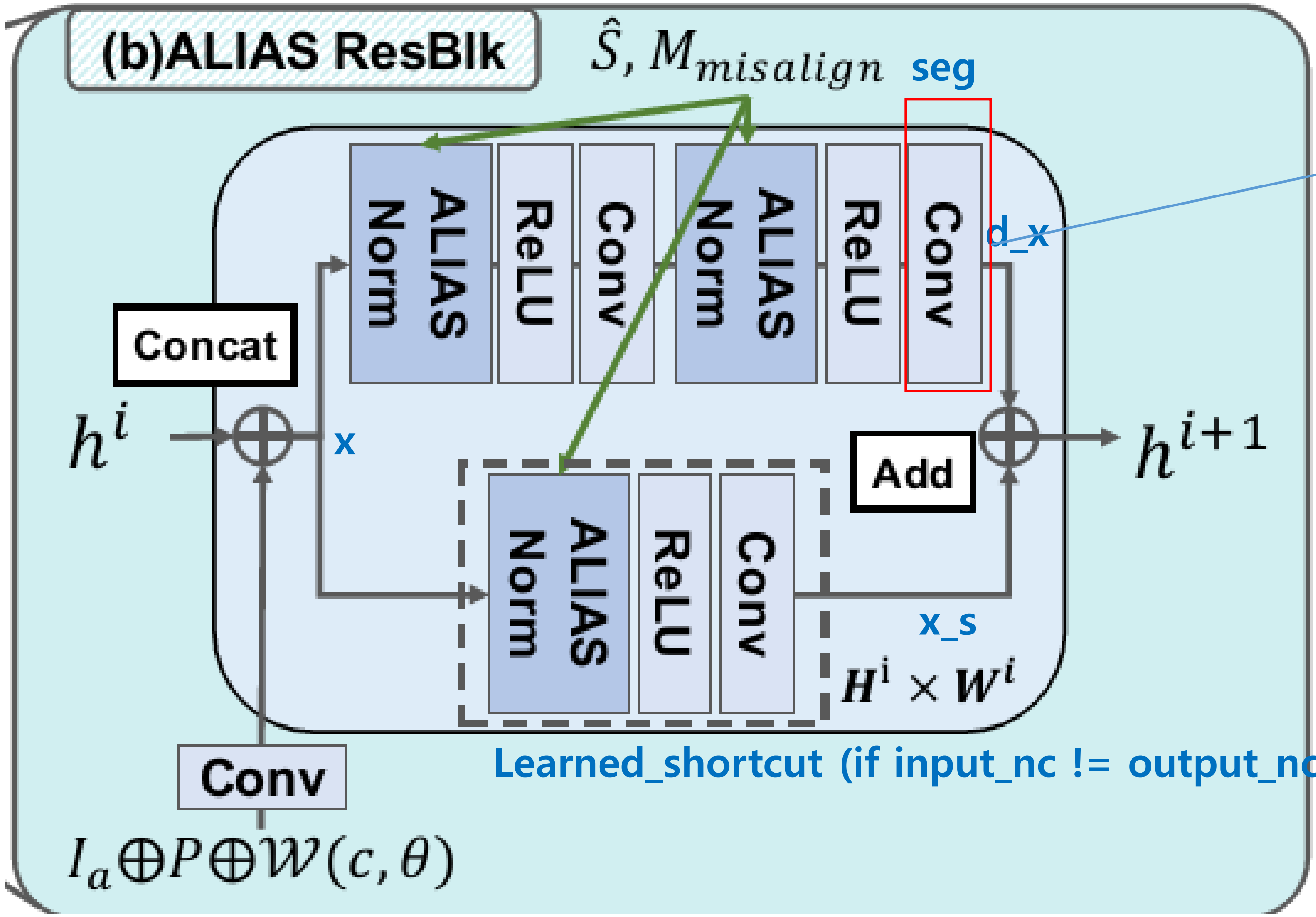
up4(1,768,1024,32)

If most: add upsample layer and resblock

conv_img(1,768,1024,3)

Multiscale refinement at feature level => preserve cloth detail

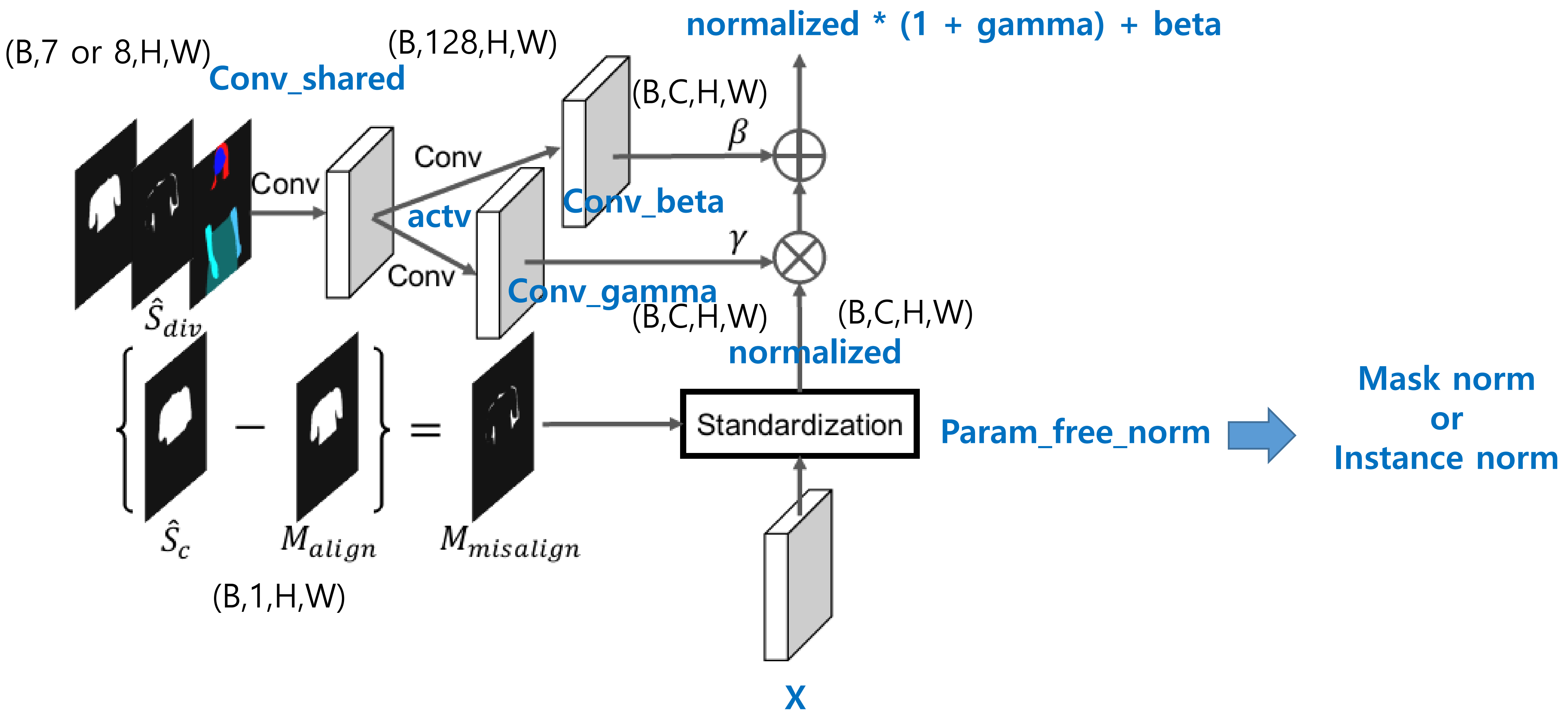
ALIAS ResBlk



Spectral normalize된 conv

Learned_shortcut (if input_nc != output_nc) => skip connection(dim 맞춰줘야)

ALIAS normalization



ALIAS normalization

Effectiveness of the ALIAS Normalization. We study the effectiveness of ALIAS normalization by comparing our model to VITON-HD*, where the standardization in ALIAS normalization is replaced by channel-wise standardization, as in the original instance normalization [30]. Fig. 8 shows that ALIAS normalization has the capability to fill the misaligned areas with the target clothing texture by removing the misleading information. On the other hand, without utilizing ALIAS normalization, the artifacts are produced in the misaligned areas, because the background information in the warped clothing image is not removed as described in Section 3.4. ALIAS normalization, however, can handle the misaligned regions properly.

Figure 8: Effects of ALIAS normalization. The orange colored areas in the enlarged images indicate the misaligned regions.

ALIAS normalization은 target cloth의 texture를 보존하면서 misalign 영역에 있는 artifacts들을 제거할 수 있다.

Misleading information=artifacts

Artifacts: warped cloth의 제거되지 않은 background (근데 이게 misalign 영역에 있나?)

mask normalization

$$h_{n,k,y,x}^{i,Mask} = Mask * h_{n,k,y,x}^{i}$$

$$(y,x) \in \Omega_n^{i,Mask}$$

$$\left|\Omega_n^{i,Mask}\right| = num\ of\,(y,x)$$

$$\mu_{n,k}^{i,mask} = \frac{1}{|\Omega_n^{i,mask}|} * \sum_{(y,x)} h_{n,k,y,x}^{i}$$

$$n_{n,k,y,x}^{i,Mask} = \sqrt{\frac{|\Omega_n^{i,Mask}|}{H*W}} * Norm_{instance}\left(h_{n,k,y,x}^{i,Mask} + \mu_{n,k}^{i,Mask} * (1 - Mask)\right)$$

$$h_{n,k,y,x}^{i+1} = n_{n,k,y,x}^{i,M_{misalign}} + n_{n,k,y,x}^{i,(1-M_{misalign})}$$

Similar to instance normalization [30], the activation is standardized per channel. However, ALIAS normalization divides the activation in channel k into the activation in the misaligned region and the other region.

Misalign 영역과 아닌 영역을 나눠서 normalize

Misalign 영역이 작을 수록 표준편차가 작음, 따라서 normalize된 값이 커짐.

1-Misalign 과 범위를 어느정도 맞추기 위해 계수를 곱하는 것 같음

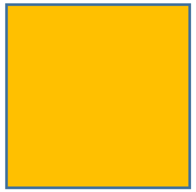
mask normalization example

$h^{i}_{n,k,y,x}$

$h^{i}_{1,1,y,x}$

1	2	3	2	2
2	2	2	2	2
1	1	3	2	3
2	2	1	2	3
3	1	2	2	2

$M_{misalign}$



$M_{misalign}$

0	0	0	0	0
0	0	0	0	0
0	1	1	1	0
0	0	0	0	0
0	0	0	0	0

$h^{i,M_{misalign}}_{1,1,y,x}$

0	0	0	0	0
0	0	0	0	0
0	1	3	2	0
0	0	0	0	0
0	0	0	0	0

2	2	2	2	2
2	2	2	2	2
2	1	3	2	2
2	2	2	2	2
2	2	2	2	2

$$h^{i,M_{misalign}}_{1,1,y,x} + (1 - M_{misalign}) * \mu^{i,M_{misalign}}_{1,1,y,x}$$

$h^{i,(1-M_{misalign})}_{1,1,y,x}$

1	2	3	2	2
2	2	2	2	2
1	0	0	0	3
2	2	1	2	3
3	1	2	2	2

1	2	3	2	2
2	2	2	2	2
1	2	2	2	3
2	2	1	2	3
3	1	2	2	2

$$h^{i,(1-M_{misalign})}_{1,1,y,x} + M_{misalign} * \mu^{i,(1-M_{misalign})}_{1,1,y,x}$$

$n^{i,M_{misalign}}_{1,1,y,x}$

$\sqrt{\frac{3}{25}} *$

0	0	0	0	0
0	0	0	0	0
0	-3.5	3.5	0	0
0	0	0	0	0
0	0	0	0	0

$\oplus$

$\sqrt{\frac{22}{25}} *$

-1.7	0	1.7	0	0
0	0	0	0	0
-1.7	0	0	0	1.7
0	0	-1.7	0	1.7
1.7	-1.7	0	0	0

$n^{i,(1-M_{misalign})}_{1,1,y,x}$

Spectral norm

$$\mathbf{W}_{SN} = \frac{\mathbf{W}}{\sigma(\mathbf{W})}, \sigma(\mathbf{W}) = \max_{\mathbf{h}:\mathbf{h}\neq 0} \frac{\|\mathbf{W}\mathbf{h}\|_2}{\|\mathbf{h}\|_2}$$

필터 정규화 -> loss function의 gradient normalize

Spectral normalization stabilizes the training of discriminators (critics) in Generative Adversarial Networks (GANs) by rescaling the weight tensor with spectral norm σ of the weight matrix calculated using power iteration method. If the dimension of the weight tensor is greater than 2, it is reshaped to 2D in power iteration method to get spectral norm. This is implemented via a hook that calculates spectral norm and rescales weight before every `forward()` call.

Discriminators를 학습을 안정화된다고 알려짐, 근데 왜 generator에 사용했는지 모르겠음
학습할 때 도움은 될 듯

Instance normalization

$$y_{tijk} = \frac{x_{tijk} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad \mu_i = \frac{1}{HWT}\sum_{t=1}^{T}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_i^2 = \frac{1}{HWT}\sum_{t=1}^{T}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - mu_i)^2.$$

Batch Normalization

$$y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}}, \quad \mu_{ti} = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_{ti}^2 = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - mu_{ti})^2.$$

Instance Normalization

Batch가 1인 batch normalization

Network가 contrast에 의존적이지 않게 한다.(style transfer등에서)

Torch.nn.instanceNorm2d:

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

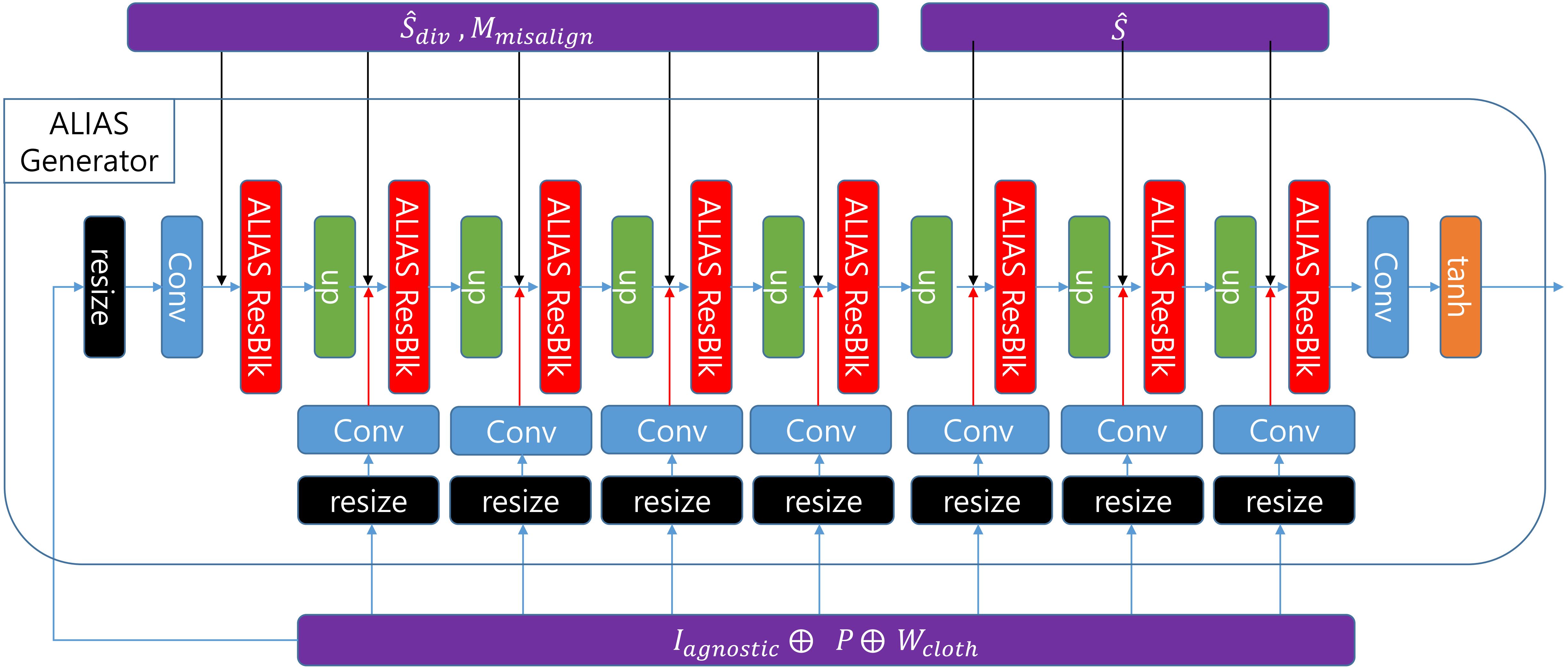
Affine=True

- Input: (N, C, H, W) or (C, H, W)
- Output: (N, C, H, W) or (C, H, W) (same shape as input)

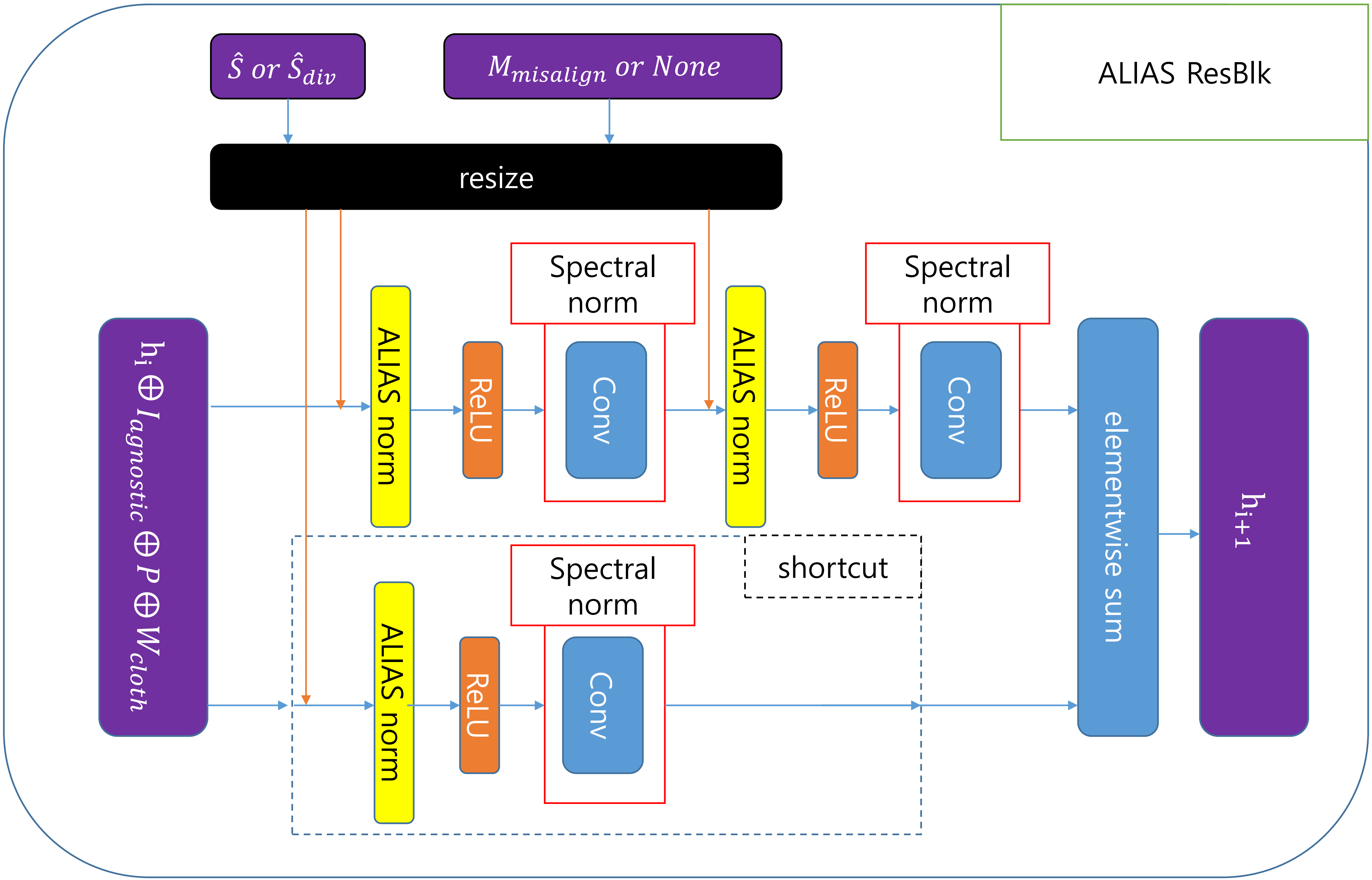
$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}}$$

Affine=False

Affine이 false이므로 적용 affine 파라미터는 무시
Style transfer에서 사용된 방식이므로 affine parameter는 dim=C인 vector

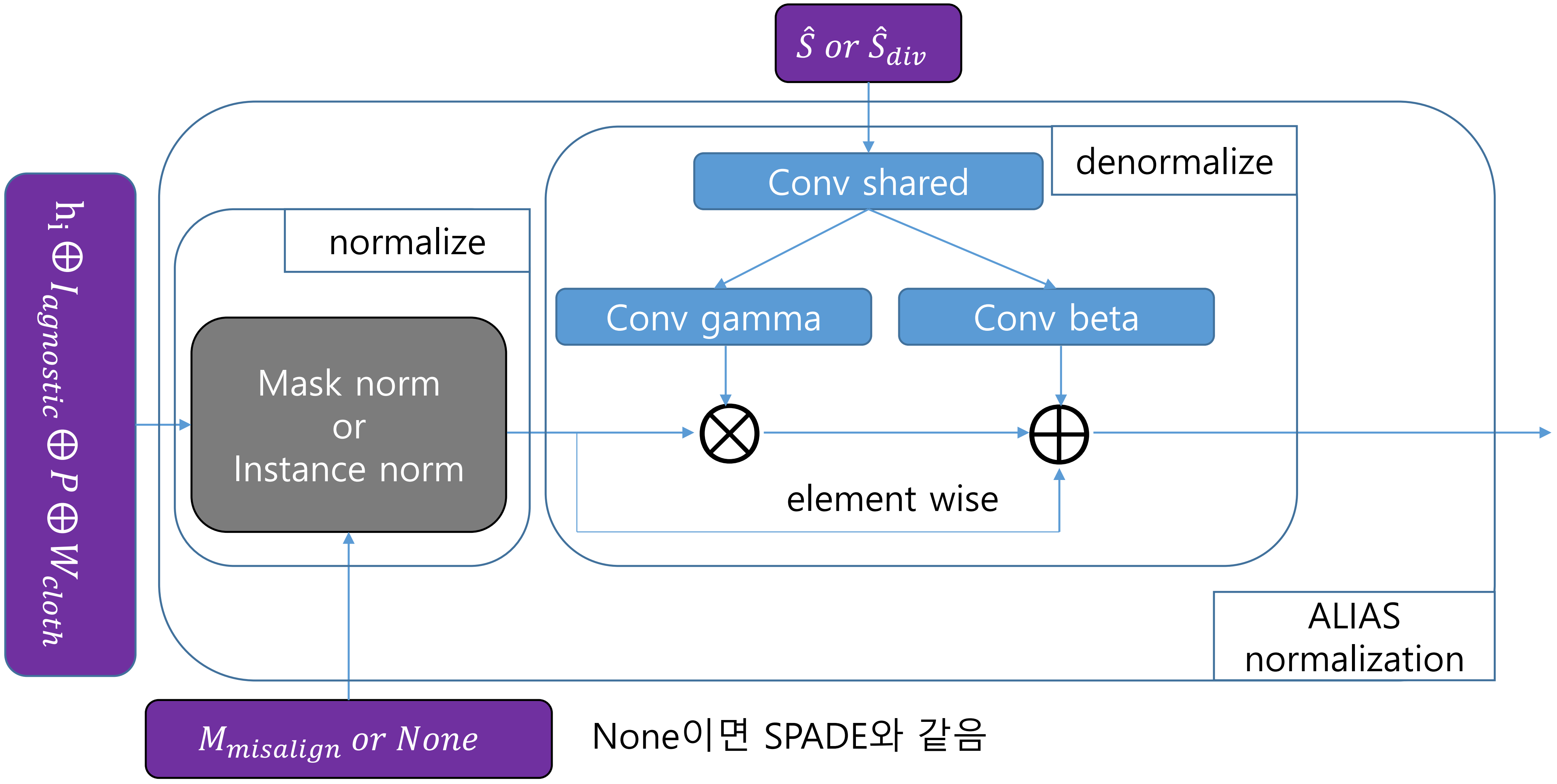


$\hat{S}_{div}, M_{misalign}$
$\hat{S}$
ALIAS Generator
resize
Conv
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
up
ALIAS ResBlk
Conv
tanh
Conv
Conv
Conv
Conv
Conv
Conv
Conv
resize
resize
resize
resize
resize
resize
resize
$I_{agnostic} \oplus P \oplus W_{cloth}$



ALIAS ResBlk
$\hat{S}$ or $\hat{S}_{div}$
$M_{misalign}$ or $None$
resize
$h_i \oplus I_{agnostic} \oplus P \oplus W_{cloth}$
ALIAS norm
ReLU
Spectral norm
Conv
ALIAS norm
ReLU
Spectral norm
Conv
shortcut
ALIAS norm
ReLU
Spectral norm
Conv
elementwise sum
h_{i+1}

ALIAS normalization



None이면 SPADE와 같음