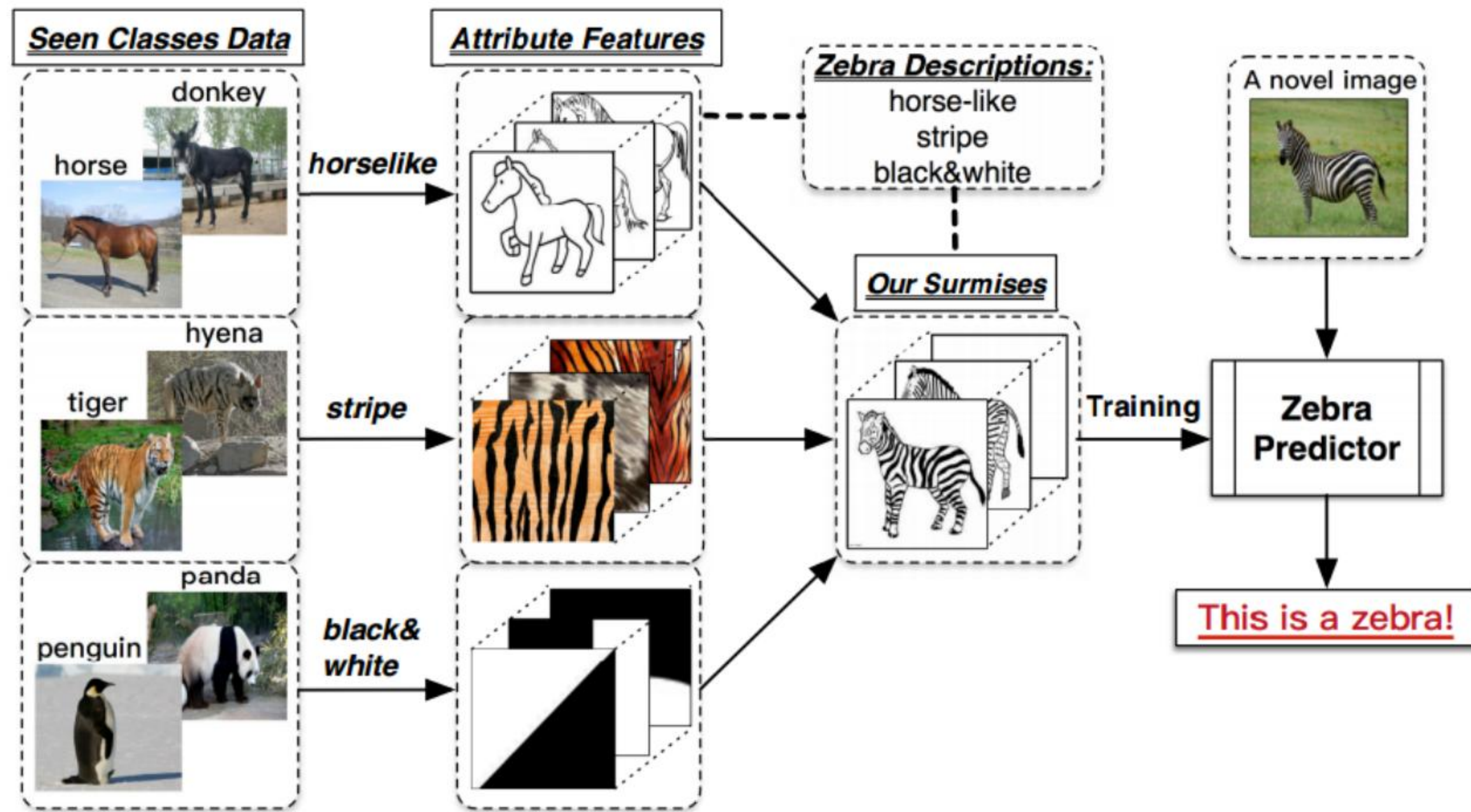


Decomposed Soft Prompt Guided Fusion Enhancing for Compositional Zero-Shot Learning

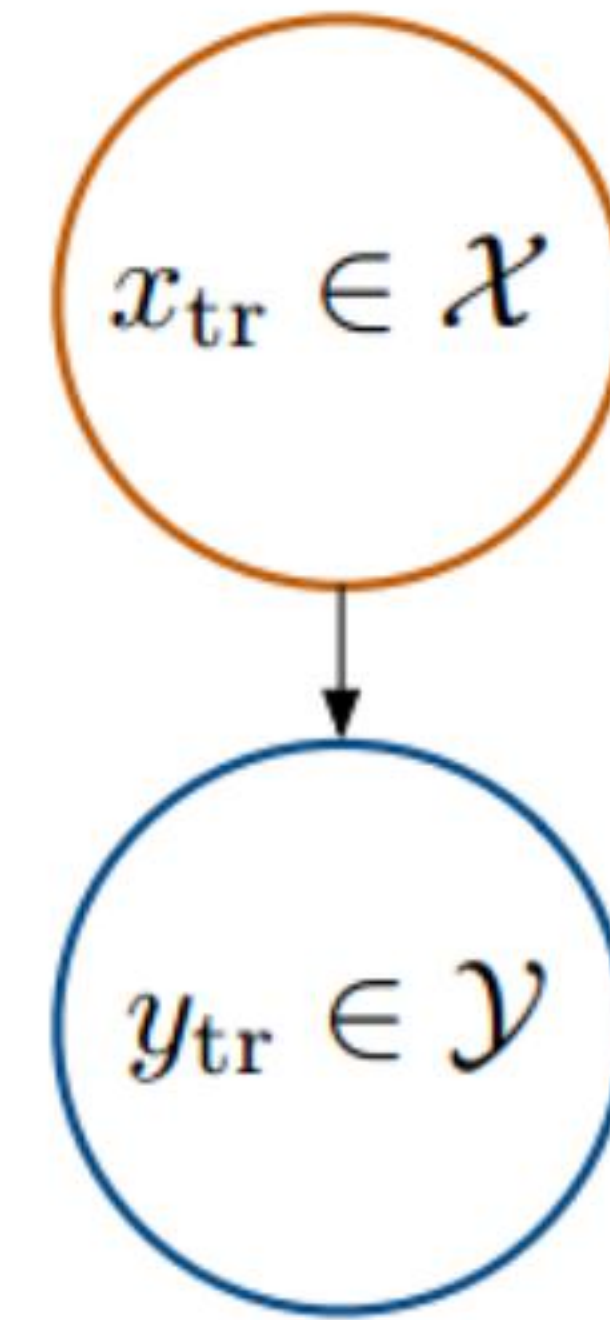
Xiaocheng Lu, Song Guo, Ziming Liu, Jingcai Guo

**THU-PM-282
Poster No. 282**

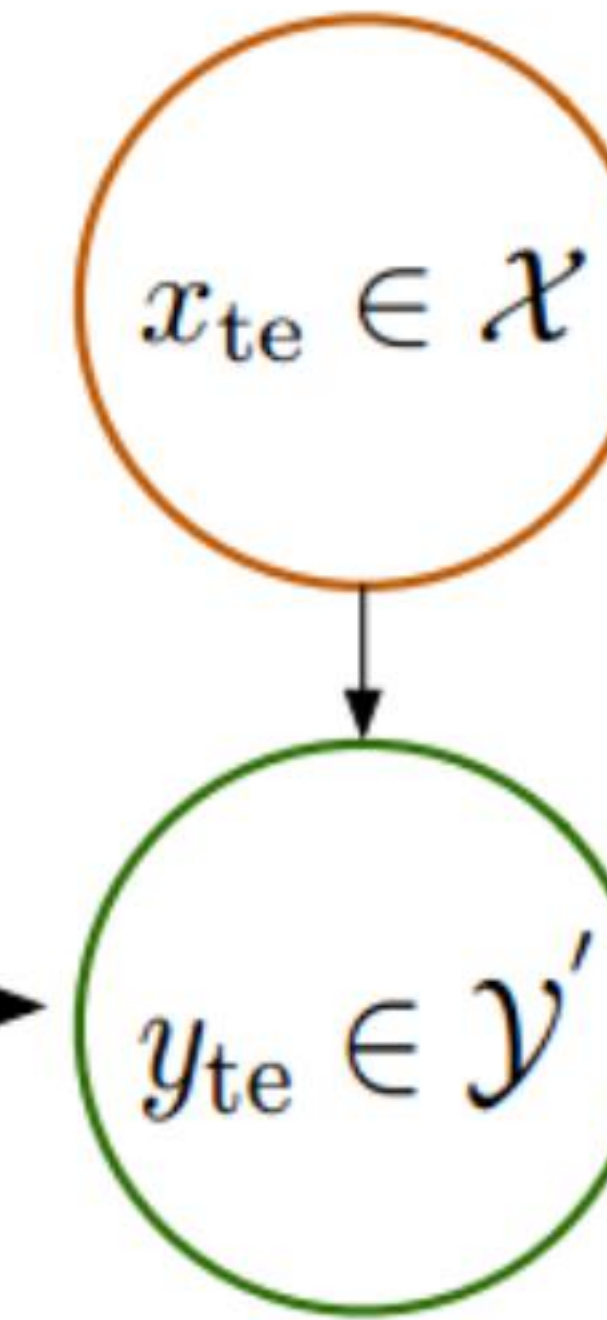
Traditional Zero-Shot Learning



Train:



Test:



Objective:

$$f: \mathcal{X} \rightarrow \mathcal{Y}'$$

$$\mathcal{Y} \cap \mathcal{Y}' = \emptyset$$

Knowledge transfer

Compositional Zero-Shot Learning

Training Stage



young tiger



old cat



Inference Stage



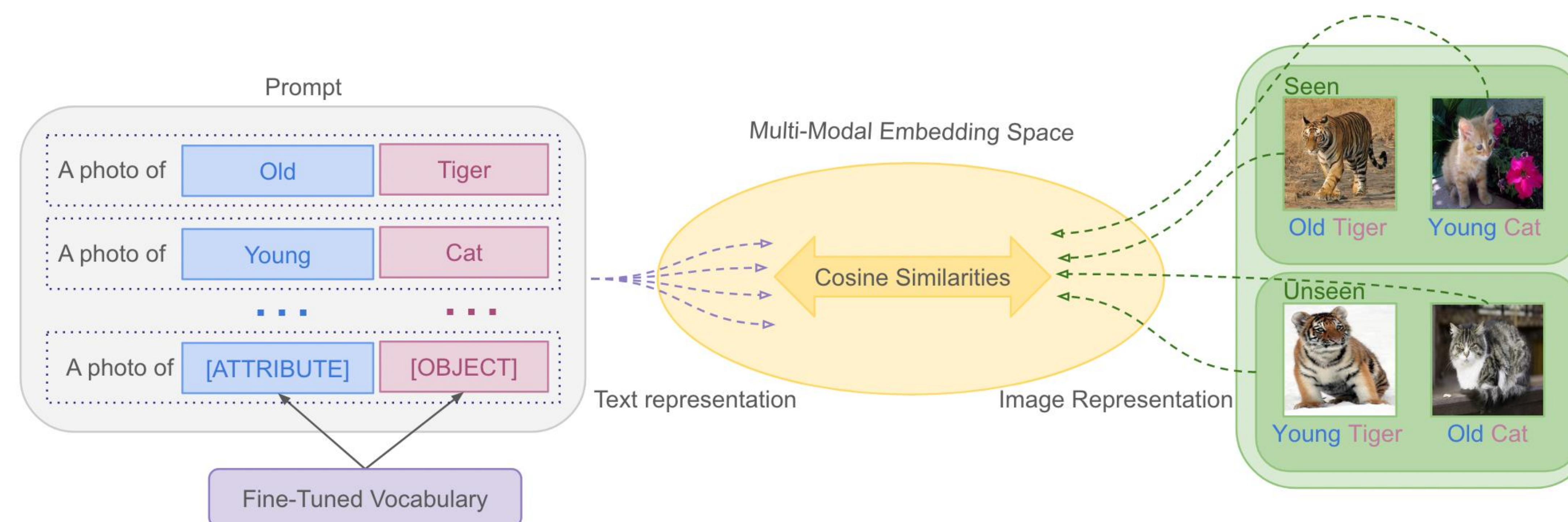
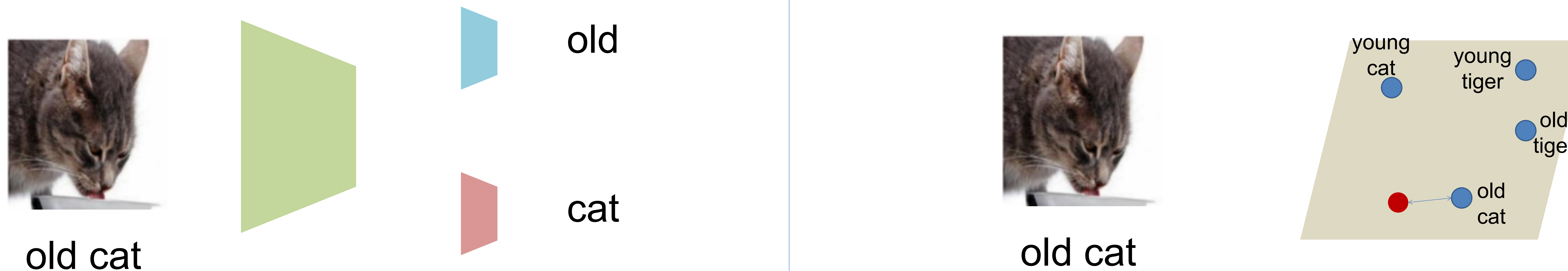
young cat



old tiger

Given *young tiger* and *old cat* in training stage, CZSL aims to identify *young cat* and *old tiger* in testing stage.

Previous Methods



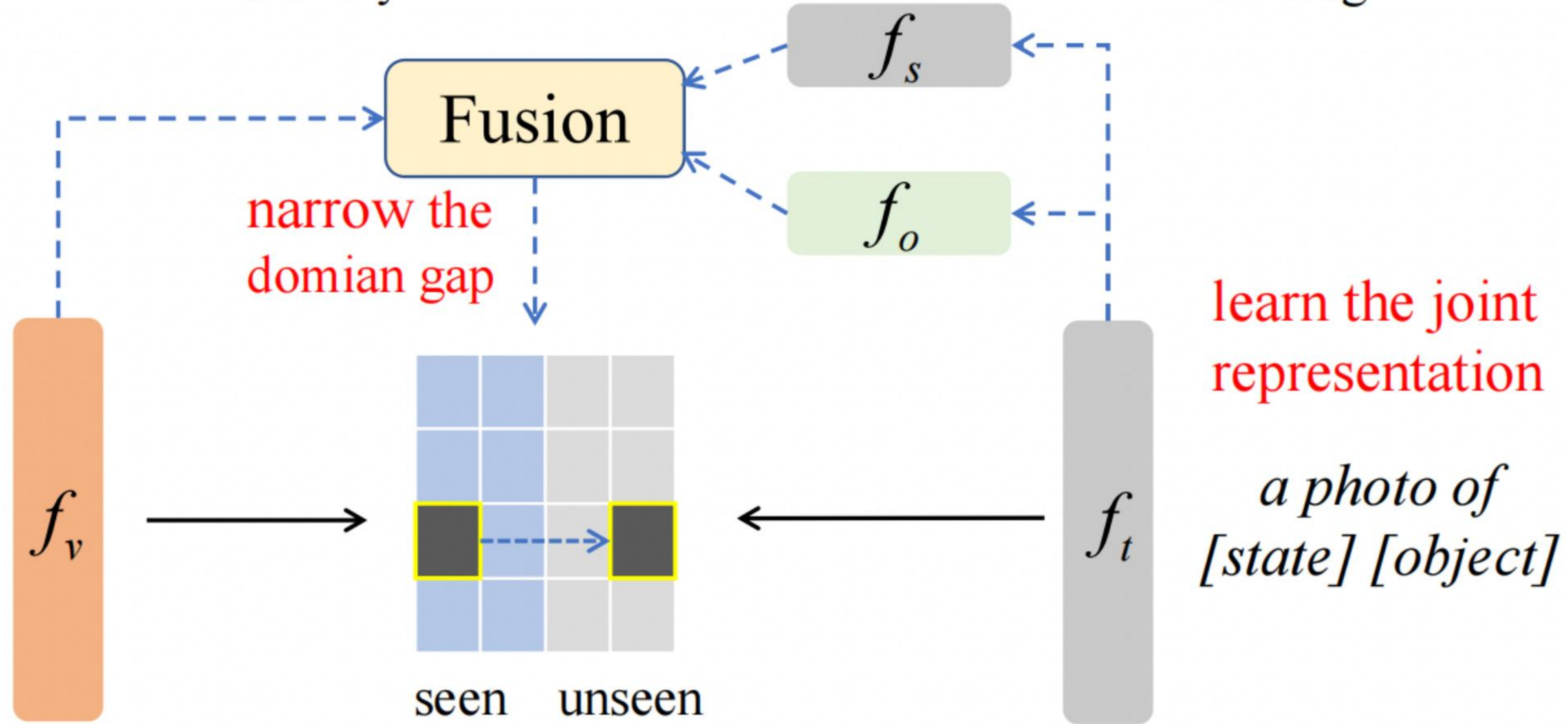
Our Motivation

$$f_v \times f_s = \begin{array}{c} \uparrow \\ \text{old} \quad \text{dry} \\ \text{old} \quad \text{dry} \end{array} \quad f_v \times f_o = \begin{array}{c} \uparrow \\ \text{cat} \quad \text{dog} \\ \text{cat} \quad \text{dog} \end{array}$$

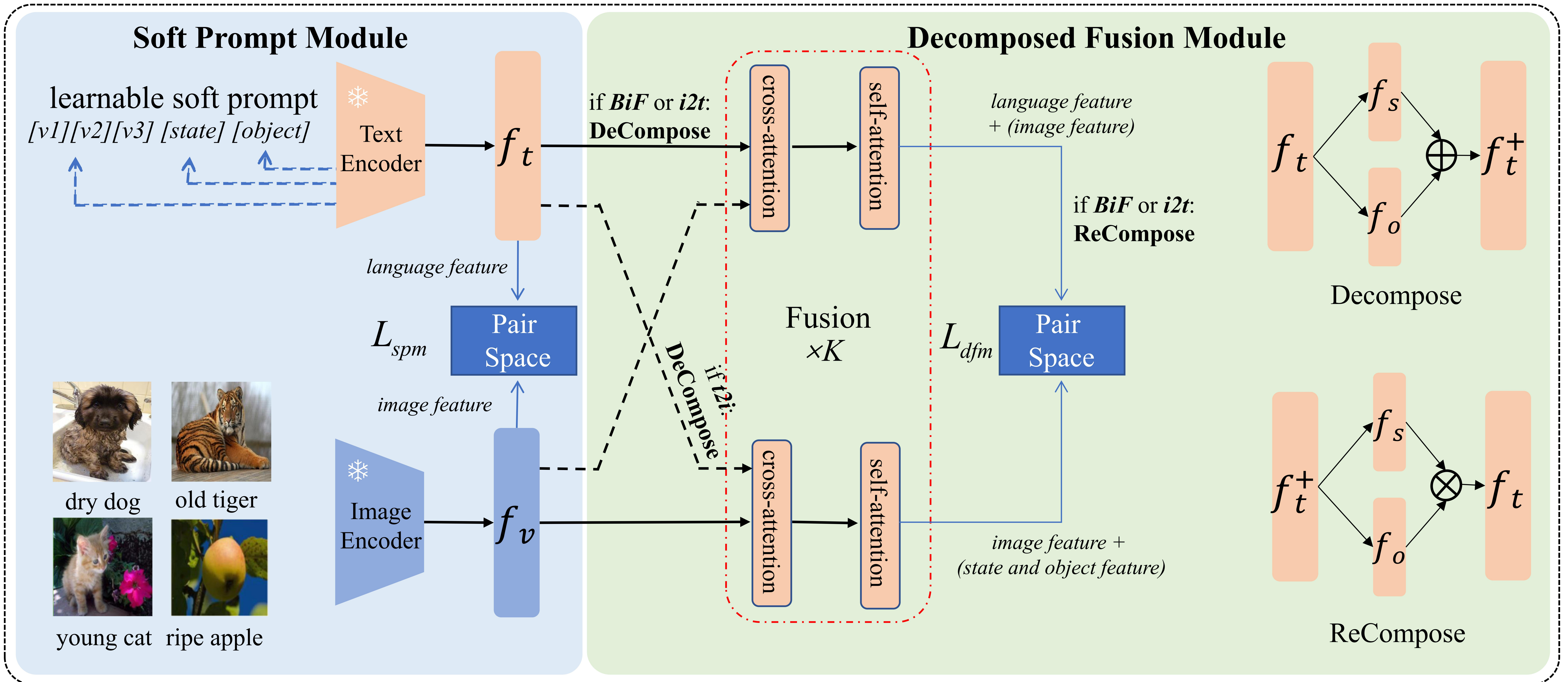
DFSP



old cat
(unseen)



Framework of DFSP



Dataset	Train		Validation			Test				
	s	o	c_s	i	c_s	c_u	i	c_s	c_u	i
MIT-States	115	245	1262	30338	300	300	10420	400	400	12995
UT-Zappos	16	12	83	22998	15	15	3214	18	18	2914
CGQA	453	870	6963	26920	1173	1368	7280	1022	1047	5098

Closed-World

Method	MIT-States				UT-Zappos				CGQA			
	S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
AoP [28]	14.3	17.4	9.9	1.6	59.8	54.2	40.8	25.9	17.0	5.6	5.9	0.7
LE+ [27]	15.0	20.1	10.7	2.0	53.0	61.9	41.0	25.7	18.1	5.6	6.1	0.8
TMN [32]	20.2	20.1	13.0	2.9	58.7	60.0	45.0	29.3	23.1	6.5	7.5	1.1
SymNet [20]	24.2	25.2	16.1	3.0	49.8	57.4	40.4	23.4	26.8	10.3	11.0	2.1
CompCos [23]	25.3	24.6	16.4	4.5	59.8	62.5	43.1	28.1	28.1	11.2	12.4	2.6
CGE [27]	28.7	25.3	17.2	5.1	56.8	63.6	41.2	26.4	28.7	25.3	17.2	5.1
Co-CGE [24]	31.1	5.8	6.4	1.1	62.0	44.3	40.3	23.1	32.1	2.0	3.4	0.5
SCEN [18]	29.9	25.2	18.4	5.3	63.5	63.1	47.8	32.0	28.9	25.4	17.5	5.5
CSP [30]	46.6	49.9	36.3	19.4	64.2	66.2	46.6	33.0	28.8	26.8	20.5	6.2
DFSP(<i>i2i</i>)	47.4	52.4	37.2	20.7	64.2	66.4	45.1	32.1	35.6	29.3	24.3	8.7
DFSP(<i>BiF</i>)	47.1	52.8	37.7	20.8	63.3	69.2	47.1	33.5	36.5	32.0	26.2	9.9
DFSP(<i>t2i</i>)	46.9	52.0	37.3	20.6	66.7	71.7	47.2	36.0	38.2	32.0	27.1	10.5

Open-World

Method	MIT-States				UT-Zappos				CGQA			
	S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
AoP [28]	16.6	5.7	4.7	0.7	50.9	34.2	29.4	13.7	-	-	-	-
LE+ [27]	14.2	2.5	2.7	0.3	60.4	36.5	30.5	16.3	19.2	0.7	1.0	0.08
TMN [32]	12.6	0.9	1.2	0.1	55.9	18.1	21.7	8.4	-	-	-	-
SymNet [20]	21.4	7.0	5.8	0.8	53.3	44.6	34.5	18.5	26.7	2.2	3.3	0.43
CompCos [23]	25.4	10.0	8.9	1.6	59.3	46.8	36.9	21.3	-	-	-	-
CGE [27]	32.4	5.1	6.0	1.0	61.7	47.7	39.0	23.1	32.7	1.8	2.9	0.47
Co-CGE^Closed [24]	31.1	5.8	6.4	1.1	62.0	44.3	40.3	23.1	32.1	2.0	3.4	0.53
Co-CGE^Open [24]	30.3	11.2	10.7	2.3	61.2	45.8	40.8	23.3	32.1	3.0	4.8	0.78
KG-SP [14]	28.4	7.5	7.4	1.3	61.8	52.1	42.3	26.5	31.5	2.9	4.7	0.78
CSP [30]	46.3	15.7	17.4	5.7	64.1	44.1	38.9	22.7	28.7	5.2	6.9	1.20
DFSP(<i>i2i</i>)	47.2	18.2	19.1	6.7	64.3	53.8	41.2	26.4	35.6	6.5	9.0	1.95
DFSP(<i>BiF</i>)	47.1	18.1	19.2	6.7	63.5	57.2	42.7	27.6	36.4	7.6	10.6	2.39
DFSP(<i>t2i</i>)	47.5	18.5	19.3	6.8	66.8	60.0	44.0	30.3	38.3	7.2	10.4	2.40

```



1 def decompose(text_feature , pair_idx):
2     t, l, c = text_feature.shape
3     att_idx, obj_idx = pair_idx[:, 0].cpu().numpy(), pair_idx[:, 1].cpu().numpy()
4     text_att = torch.zeros(t, self.attributes, c).cuda()
5     text_obj = torch.zeros(t, self.classes, c).cuda()
6     for i in range(self.attributes):
7         text_att[:, i, :] = text_feature[:, np.where(att_idx==i)[0], :].mean(-2)
8     for i in range(self.classes):
9         text_obj[:, i, :] = text_feature[:, np.where(obj_idx==i)[0], :].mean(-2)
10    text_feature_plus = torch.cat([text_att, text_obj], dim=1)
11    return text_feature_plus
12
13
14 def recompose(text_feature_plus , pair_idx):
15    t, l, c = text_feature_plus.shape
16    att_idx, obj_idx = pair_idx[:, 0].cpu().numpy(), pair_idx[:, 1].cpu().numpy()
17    text_com_feature = torch.zeros(t, len(idx), c).cuda()
18    text_com_feature = text_feature_plus[:, att_idx, :] * text_feature_plus[:, obj_idx + offset, :]
19    return text_com_feature
  
```

Code 1: Decompose and Recompose

Success Cases

<p>ancient castle</p>  <p>ancient castle old castle huge castle</p>	<p>ancient city</p>  <p>ancient city ancient buliding eroded desert</p>	<p>barren field</p>  <p>barren field empty field narrow field</p>
<p>bent knife</p>  <p>bent knife bent blade bent handle</p>	<p>bright cloud</p>  <p>bright cloud cloudy cloud windblown cloud</p>	<p>broken branch</p>  <p>broken branch broken log splintered tree</p>
<p>broken door</p>  <p>broken door splintered door weathered door</p>	<p>moldy bread</p>  <p>moldy bread moldy cheese moldy fruit</p>	<p>old church</p>  <p>old church ancient church small church</p>
<p>broken camera</p>  <p>broken camera scratched camera broken mirror</p>	<p>fresh fruit</p>  <p>fresh fruit fresh berry diced fruit</p>	<p>wrinkled leaf</p>  <p>wrinkled leaf crinkled leaf ruffled leaf</p>

Failure Cases

<p>small island</p>  <p>tiny island small island huge island</p>	<p>cracked door</p>  <p>broken door cracked door broken window</p>
<p>modern library</p>  <p>large library huge library small library</p>	<p>spilled oil</p>  <p>weathered oil spilled oil burnt oil</p>
<p>crinkled silk</p>  <p>draped dress crinkled dress creased dress</p>	<p>large bottle</p>  <p>cut glass small bottle engraved glass</p>
<p>rusty car</p>  <p>rusty truck rusty car old car</p>	<p>cooked fish</p>  <p>thawed salmon thawed fish browned pie</p>