Appendix

- In Sec. A1, we introduce more details about the structure of SAPG and provide several examples of the object classes and the set of prompts that are used in our experiments.
- In Sec. A2, we present a more in-depth explanation of this paper's motivation. Specifically, we compute statistics on the presence of symmetry cues within the vision-language dataset, analyze the benefits of language in symmetry detection from a theoretical perspective, and further discuss why SAPG works better.
- In Sec. A3, we present additional ablation studies and visualization results.
- In Sec. A4, we provide additional implementation details.

A1. Structure of text prompts

To construct language prompts for symmetry detection, we use Grounded-SAM [44] to extract the frequent 2081 object classes from the DENDI dataset. The full list will be provided in the released code. Below we show the first 100 objects.

man, pole, stand, white, building, sit, table, floor, sky, person, red, street sign, food, traffic sign, road, clock, plate, green, attach, catch, sign, park, peak, street corner, tree, platter, woman, car, stop sign, blue, tower, black, play, lush, blanket, yellow, road sign, stool, bell tower, grass, curb, tray, field, walk, stare, cloudy, pavement, ball, child, dinning table, photo, water, boy, ride, spire, animal, girl, drive, brown, fill, vegetable, cat, fly, footstall, room, hand, sea, lay, cup, container, pillar, flower, city, beverage, motorcycle, grassy, bowl, license plate, wear, fruit, shirt, countertop, dog, snow, plane, lamp, rail, motorbike, home appliance, toy, stone building, electronic, bus, chair, swinge, pizza, racket, tennis racket, rural, vase

Therefore, if we want to construct a set of prompts $\mathcal{T}$ with $M = 3$ prompts and each containing $K = 3$ words, considering using the first $3 \times 3 = 9$ objects as an example, $\mathcal{T}$ can be construct as follows:

$$\mathcal{T} = \{\underbrace{\textit{"man pole stand"}}_{t_1}, \underbrace{\textit{"white building sit"}}_{t_2}, \underbrace{\textit{"table floor sky"}}_{t_3}\}$$

Note, the "frequent objects ($K = 25$)" row in Tab. 5 uses the first 25 objects in the above list.

As stated in Sec. 3.2, during training, the prompts are fixed and shared across all images. Our goal is not to search for the optimal prompt because the searching space is extremely large, but to show the grouping structure is helpful. We further discuss the benefits of this design in Sec. A2.3.

Shape/Symmetry Word	**Occurrence** (%)
Ring	4.2718
Line	1.9806
Arc	1.5185
Ball	1.4913
Square	0.4095
Oval	0.1699
Cone	0.1606
Arrow	0.1572
Circle	0.1518
Globe	0.0892
Rectangle	0.0830
Cube	0.0776
Grid	0.0708
Pyramid	0.0685
Triangle	0.0592
Spiral	0.0503
Sphere	0.0413
Cylinder	0.0324
Hexagon	0.0277
Crescent	0.0274
Prism	0.0163
Octagon	0.0109
Checkerboard	0.0070
Helix	0.0066
Pentagon	0.0050
Ellipse	0.0035
Rhombus	0.0025
Trapezoid	0.0021
Torus	0.0007
Semicircle	0.0006
Dodecahedron	0.0005
Tetrahedron	0.0005
Icosahedron	0.0004
Parallelogram	0.0004

Table A1. Percentage of image captions containing shape/symmetry related words in the LAION-400M dataset.

A2. Discussions on the impacts of language

A2.1. Language cues about symmetry in CLIP's pre-training

LAION-400M [48] is a large-scale public dataset containing $400M$ image-caption pairs, which is potentially similar to the dataset that CLIP was trained on. In Tab. A1, we use GPT-4o to generate a few symmetry and shape-related words and calculate the percentage of the occurrence of these words within LAION-400M's captions. We observe that common shape-associated words such as 'ring,' 'line,' and 'ball' have appeared more frequently than complex geometric shapes such as 'parallelogram,' 'icosahedron,' and 'tetrahedron.' Nevertheless, as the dataset is very large, even the occurrence of 0.0004% translates to the presence of 1600 image-caption pair containing complex shape concepts such as 'icosahedron.' Pre-training on such diverse image-caption pairs enables the CLIP model to learn image representations that encode rich symmetry-related information.

A2.2. A theoretical perspective on the benefits of language

In this subsection, we provide a theoretical perspective to analyze the benefits of using language in the symmetry detection task.

Hypothesis 1. *Suppose there exists a perfect image encoder $E^*_{\texttt{img}}$ which leads to the best visual features for image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, i.e., $\mathbf{Z}^*_{\boldsymbol{I}} = E^*_{\texttt{img}}(\boldsymbol{I})$. Provided that language contains cues about symmetry, we assume the best visual features are offset by an additive term $\delta^*(t)$ that depends on a language prompt t, plus zero-mean noise $\varepsilon_{\boldsymbol{I}}$:*

$$\mathbf{Z}_{\boldsymbol{I}} = \mathbf{Z}^*_{\boldsymbol{I}} - \delta^*(t) + \varepsilon_{\boldsymbol{I}}, \tag{A19}$$

where $\mathbb{E}[\varepsilon_{\boldsymbol{I}}] = 0$ and $\delta^*(t) \neq 0$ when language provides symmetry cues.

Then we make a claim that language is beneficial under Hypothesis 1:

Claim 2. *Using language prompt t, a FiLM layer of the form (following equation 3)*

$$\mathbf{Z}_{\boldsymbol{I}|t} = \gamma(z_t) \odot \mathbf{Z}_{\boldsymbol{I}} + \beta(z_t), \tag{A20}$$

with elementwise multiplication $\odot(\cdot)$ and trainable linear mappings $\gamma(\cdot), \beta(\cdot) \in \mathbb{R}^d$, can reduce the expected error of visual features. Formally,

$$\mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}|t} - \mathbf{Z}^*_{\boldsymbol{I}}\|\big] < \mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}} - \mathbf{Z}^*_{\boldsymbol{I}}\|\big] \tag{A21}$$

Proof. According to A19, the ideal additive fix to $\mathbf{Z}_{\boldsymbol{I}}$ is

$$f^*(\mathbf{Z}_{\boldsymbol{I}}) = \mathbf{Z}_{\boldsymbol{I}} + \delta^*(t), \tag{A22}$$

where $f(\cdot)$ represents a function which modulates $\mathbf{Z}_{\boldsymbol{I}}$ to fit for the symmetry detection task, and $f^*(\cdot)$ correpsondingly represents the best fix function.

We then show FiLM can implement $f*(\cdot)$. Simply choose

$$\gamma(z_t) = \mathbf{1}(\text{all ones vector}), \quad \beta(z_t) = \delta^*(t) \tag{A23}$$

and apply to A20, then

$$\mathbf{Z}_{\boldsymbol{I}|t} = \mathbf{Z}_{\boldsymbol{I}} + \delta^*(t) = \mathbf{Z}^*_{\boldsymbol{I}} + \varepsilon_{\boldsymbol{I}}. \tag{A24}$$

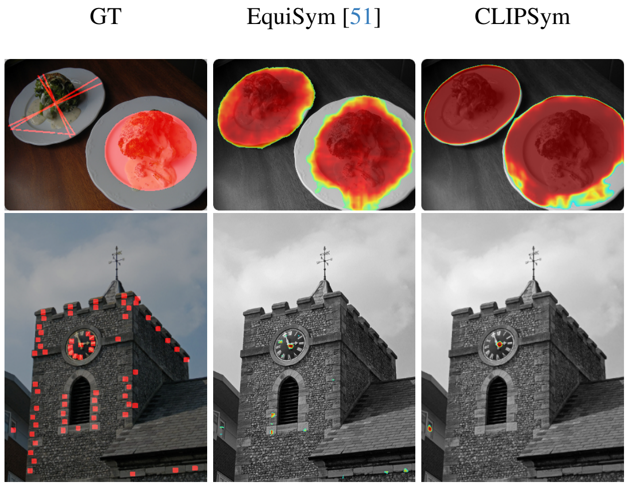


Figure A1. Examples of when symmeries cannot be well detected. The top row corresponds to the reflection case, and the bottom row corresponds to the rotation case.

Hence, the language modulated visual features differ from the best visual features $\mathbf{Z}^*_{\boldsymbol{I}}$ by only $\varepsilon_{\boldsymbol{I}}$ rather than $\delta^*(t) - \varepsilon_{\boldsymbol{I}}$. Therefore,

$$\mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}|t} - \mathbf{Z}^*_{\boldsymbol{I}}\|^2\big] \quad = \mathbb{E}\big[\|\varepsilon_{\boldsymbol{I}}\|^2\big], \tag{A25}$$

while

$$\begin{aligned}
\mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}} - \mathbf{Z}^*_{\boldsymbol{I}}\|^2\big] &= \mathbb{E}\big[\|\delta^*(t) - \varepsilon_{\boldsymbol{I}}\|^2\big] && \text{(A26)}\\
&= \mathbb{E}\big[\|\varepsilon_{\boldsymbol{I}}\|^2\big] + \mathbb{E}\big[\|\delta^*(t)\|^2\big] - 2\mathbb{E}\big[\varepsilon_{\boldsymbol{I}}\delta^*(t)\big] && \text{(A27)}\\
&= \mathbb{E}\big[\|\varepsilon_{\boldsymbol{I}}\|^2\big] + \mathbb{E}\big[\|\delta^*(t)\|^2\big] && \text{(A28)}\\
&> \mathbb{E}\big[\|\varepsilon_{\boldsymbol{I}}\|^2\big], && \text{(A29)}
\end{aligned}$$

which proves

$$\mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}|t} - \mathbf{Z}^*_{\boldsymbol{I}}\|\big] < \mathbb{E}\big[\|\mathbf{Z}_{\boldsymbol{I}} - \mathbf{Z}^*_{\boldsymbol{I}}\|\big] \tag{A30}$$

The strict inequality holds whenever $\delta^*(t) \neq 0$. □

Discussion. From this proof, we see that if the language prompt t provides additional symmetry cues ($\delta^*(t) \neq 0$), then a FiLM layer can "add back" these missing cues into the visual features, reducing the overall error. Our choice $\gamma(z_t) = \mathbf{1}$ and $\beta(z_t) = \delta^*(t)$ is simply a constructive example illustrating FiLM's ability to perform an additive correction. In practice, $\gamma(\cdot)$ and $\beta(\cdot)$ are gradually learned to approximate this fix and achieve a lower error than relying on vision features alone.

A2.3. Why does SAPG work better?

Initialization. Fig. A2 shows two illustrative examples of the predicted symmetry heatmaps under different prompting strategies at the initial step, i.e., before training. We can see that using unrelated random prompts causes the model to overly focus on most pixels in the image, while it fails to concentrate on regions that likely exhibit symmetry. For instance, in the second column, since *"cat"* or *"tree"* do not exist in the original images, the model cannot find the correct focus. In the third column, we design prompts specifically corresponding to symmetric objects, e.g., *"ice cream"* and *"balloon"*. Although the model's focus on symmetric objects improves compared to using unrelated random prompts, the distinction is still not enough. In the last column, the prompt aggregation via SAPG enables the model to correctly concentrate on symmetric objects.

The underlying reason that this improved initialization is beneficial because it provides the model with a strong semantic prior derived from frequently occurring objects which are associated with symmetry cues. As a result, the model starts from a more informed state, reducing noise and misalignment in the early stages of training.

Prompt grouping. In contrast to a single prompt, which captures only one aspect of the semantic information and may suffer from high variance due to noisy or limited cues, aggregating multiple prompt-conditioned outputs via SAPG acts as an ensemble. While the predictions from individual prompts are indeed correlated due to the shared encoders, each prompt still focuses slightly different semantic cues about symmetry. Furthermore, since the aggregation weights are learnable so that the model can put more weights on prompts that are more aligned with symmetry. These factors lead to reduced noise and more stable predictions.

Why fixed prompts rather than adaptive ones? Although one may consider using adaptive prompts that vary per image, we choose fixed prompts for several reasons. First, since CLIP's language encoder is trainable, as a result, the prompt embeddings gradually evolve during training to better capture the universal concept of symmetry. Second, adaptive prompts can be difficult to learn and may not generalize well. While it might be possible to identify an optimal prompt for each training image, generating new, adaptive prompt combinations for unseen images has the risk of producing semantic cues that do not reliably represent symmetry.

In the future, a possible direction is exploring image-correlated language embeddings, which has the potential to better represent more refined cues of symmetry based on each image's content.

A3. Additional results

A3.1. Limitations

Fig. A1 shows some cases in the DENDI dataset when symmetry cannot be well detected. The top row corresponds to the reflection case, while the bottom row corresponds to the rotation case. The limitations are likely due to the dataset's annotation quality, such as inconsistency and ambiguity. For example, the left round object (plate) in the reflection example is not annotated as a circle as it usually should be, and the complicated rotation centers on the city wall in the rotation example are not obvious. We leave the improvement of the dataset quality for future work.

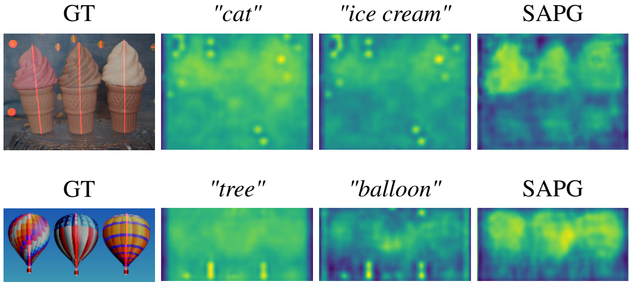


Figure A2. Illustrative examples of predicted symmetry heatmaps under different propmpting strategies at the initial step.

A3.2. More visualization results on DENDI dataset

Fig. A3 shows more qualitative results of reflection and rotation symmetry detection on the DENDI dataset. The results further show that CLIPSym can detect both reflection and rotation symmetries more effectively than other baselines.

A3.3. Consistency and robustness results on SDRW and LDRS datasets

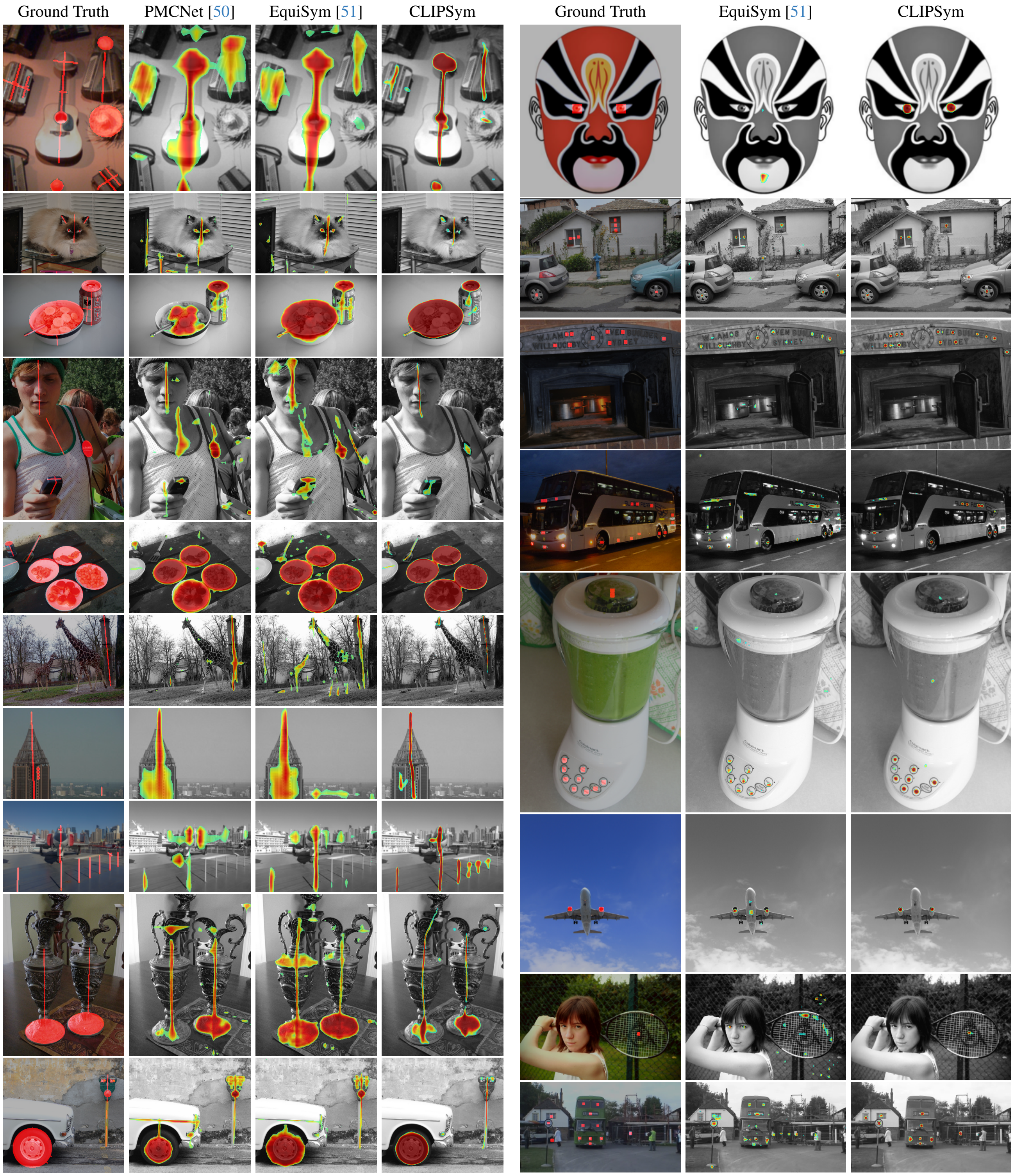
In Tab. A2, we provide the consistency and robustness evaluation results for SDRW and LDRS reflection datasets under $[-45°, 45°]$ uniformly distributed rotation operations. Similar to results in Fig. 3, CLIPSym achieves the best performance in terms of both robustness and consistency on every dataset that we evaluated.

Method	**SDRW**		**LDRS**		**Mixed**	
	Rob.↑	**Cons.↓**	**Rob.↑**	**Cons.↓**	**Rob.↑**	**Cons.↓**
PMCNet [50]	40.4	0.263	21.6	0.356	25.0	0.333
EquiSym [51]	39.4	0.101	20.5	0.112	24.8	0.109
CLIPSym$^{\text{non-eq.}}$	42.2	0.059	27.3	0.061	30.2	0.060
CLIPSym$^{\text{eq.}}$	**44.3**	**0.042**	**29.2**	**0.042**	**32.3**	**0.042**

Table A2. Equivariance robustness and consistency evaluation results for SDRW and LDRS reflection datasets under $[-45°, 45°]$ uniformly distributed rotation operations.

A3.4. Visualizations of robustness and consistency

In Fig. 3, we present heatmaps of EquiSym and CLIPSym which take images under random rotation transformations within $[-45°, 45°]$ as inputs to illustrate the model's robustness and consistency. In Fig. A4, we present more results in addition to Fig. 3.



(a) Reflection detection results on DENDI-*ref*.

(b) Rotation detection results on DENDI-*rot*.

Figure A3. Visualization of the reflection and rotation symmetry detection on the DENDI dataset.

A3.5. Ablation study on different equivariance degrees

In Tab. A3, we evaluate the F1-score on DENDI reflection dataset under different degrees of group-equivariance in the

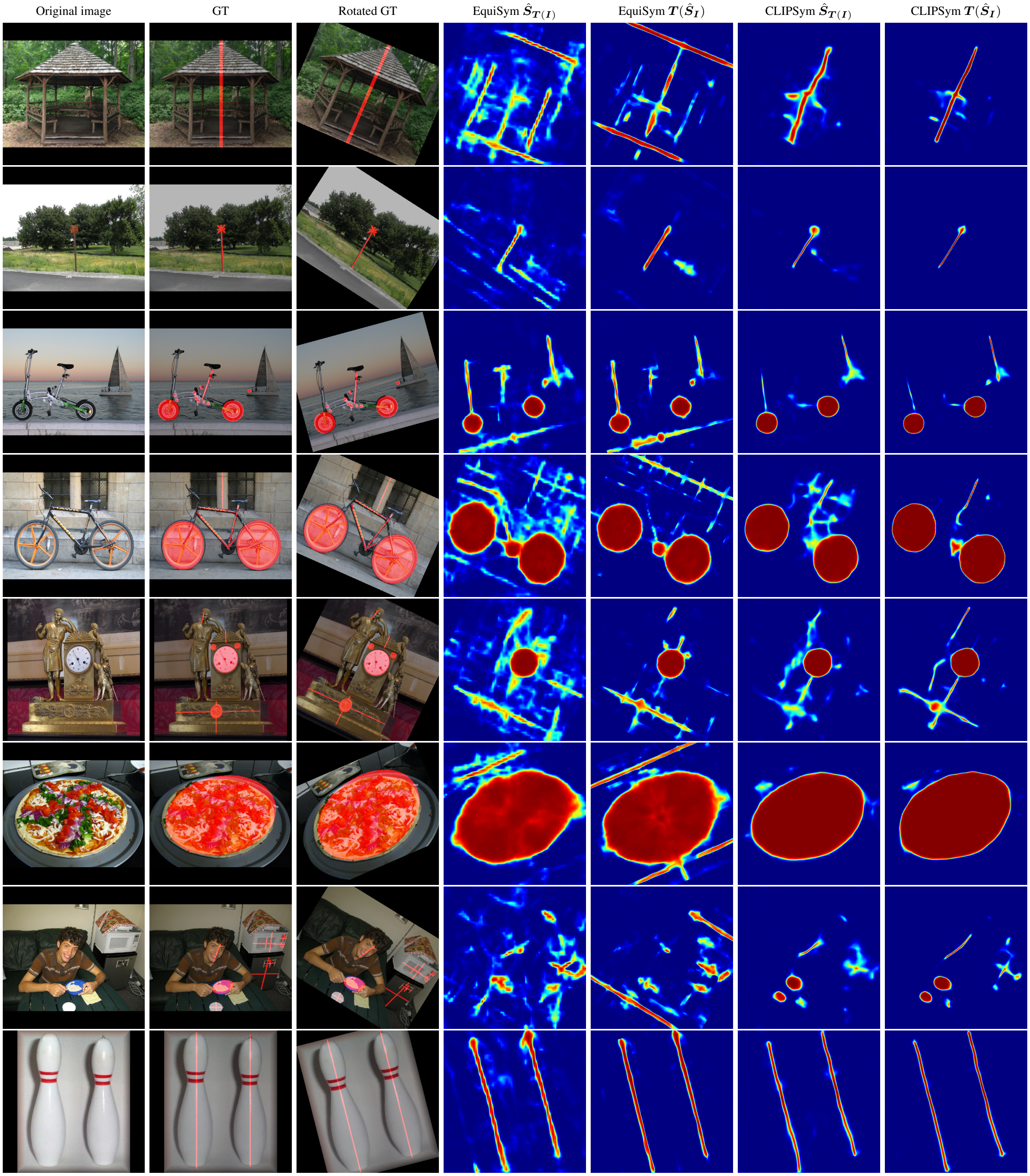

Original image
GT
Rotated GT
EquiSym $\hat{\boldsymbol{S}}_{\boldsymbol{T}(\boldsymbol{I})}$
EquiSym $\boldsymbol{T}(\hat{\boldsymbol{S}}_{\boldsymbol{I}})$
CLIPSym $\hat{\boldsymbol{S}}_{\boldsymbol{T}(\boldsymbol{I})}$
CLIPSym $\boldsymbol{T}(\hat{\boldsymbol{S}}_{\boldsymbol{I}})$

Figure A4. Examples of original image, ground truth, rotated ground truth, EquiSym and CLIPSym's predicted heatmaps $\hat{\boldsymbol{S}}_{\boldsymbol{T}(\boldsymbol{I})}$ on the rotated image and the rotated heatmap $\boldsymbol{T}(\hat{\boldsymbol{S}}_{\boldsymbol{I}})$.

design of CLIPSym decoder. Specifically, we evaluate C_4-, C_6-, C_8-, C_{12}- and C_{16}-equivariant decoders. The results show that C_8-equivariant decoder, which is the same as our model's setting, achieves the best performance.

Table A3. Quantitative comparison of F1-score (%) on the DENDI [51] reflection dataset under different degrees of group-equivariance.

Equiv. degrees	C_4	C_6	C_8	C_{12}	C_{16}
Ref. F1	65.3	64.3	**66.5**	65.6	65.8

A3.6. The best prompts structure for symmetry detection

We conduct hyperparameter search over the number of prompts $M \in \{1, 10, 25, 50\}$ and the number of object classes in each prompt $K \in \{1, 4, 8, 16, 32\}$, as well as generating different combinations of objects using different seeds, the best prompts structure in our experiments of both reflection and rotation symmetry detection has $M = 25$ and $K = 4$ and is given as Tab. A4.

A3.7. Variant of non-equivariant version of CLIPSym with 8x decoder channels

The only difference between CLIPSym's the non-equivariant version (CLIPSym$^{\text{non-eq.}}$) and the equivariant version CLIPSym$^{\text{eq.}}$ is whether the upsampler is equivariant or not. In CLIPSym$^{\text{non-eq.}}$ we use regular CNN, while in CLIPSym$^{\text{eq.}}$ we use G-convolution. Both models have the same number of channel dimensions $[64, 32, 16, 1]$ and 3×3 filters. To ensure a fair comparison to highlight the design of the equivariant module indeed helps, we also evaluate CLIPSym$^{\text{non-eq.}}$ with 8 times more channels in Tab. A5. Even with the similar decoder capacity, by comparing with results in Tab. 1 and Tab. 2, CLIPSym$^{\text{eq.}}$ outperforms this larger baseline, demonstrating the benefits of the equivariant design.

Dataset	DENDI Ref.	DENDI Rot.	SDRW	LDRS	Mixed
F1	64.9 ± 0.2	24.4 ± 0.1	49.5 ± 0.3	37.6 ± 0.2	41.4 ± 0.2

Table A5. CLIPSym$^{\text{non-eq.}}$ with 8 times more channels.

A3.8. Comparison with an emerging result

Seo and Cho [49] recently proposed to lift 2-D features into the camera's 3-D space and regress a seed vertex, which leads to a higher F1 score for rotation center detection on DENDI. However, their pipeline is restricted to rotation symmetry, which requires an extra 2D to 3D label conversion, and cannot handle reflection. In contrast, **CLIPSym** offers a *unified* solution for both reflection *and* rotation.

A4. Additional implementation details

Image processing. To conduct a fair comparison with other baselines, images with different resolutions are reshaped to 417×417 pixels as in [51] and [50] before feeding into the model. We preserve the aspect ratio of the original images and pad the shorter side with zeros. During training, data augmentations include random rotations of intervals of 90 degrees, random small rotations within $[-15°, 15°]$, and color jittering. Since the image encoder uses ViT-B/16 [7], the size of each image patch is 16×16 pixels, and the number of patches of each side of images is $M = \lfloor 417/16 \rfloor = 26$. During testing, we still follow the same image processing pipeline as in training but without data augmentations to fit the model's input size. When calculating the metrics, zero paddings are cropped and images are resized back to their original sizes.

Decoder details. The decoder's FiLM block described in Sec. 3.1 modulates the image features conditioned on text prompts. To conduct the element-wise multiplication in Eq. (3), we project the text token features and image token features to the same dimension using linear layers with learnable parameters. The dimension of the linear layer is set to $d = 64$. The Transformer module consists of $L_B = 3$ multi-headed attention layers, while the upsampler contains $3\times$ G-conv. and bi-linear interpolation submodules.

Focal loss. In the α-focal loss defined in Eq. (16), we have

$$\hat{\boldsymbol{S}}'_{\boldsymbol{I}_{xy}} = \begin{cases} \hat{\boldsymbol{S}}_{\boldsymbol{I}_{xy}} & \text{if } \boldsymbol{S}_{\boldsymbol{I}_{xy}} = 1 \\ 1 - \hat{\boldsymbol{S}}_{\boldsymbol{I}_{xy}} & \text{otherwise} \end{cases} \tag{A31}$$

$$\alpha'_{\boldsymbol{I}_{xy}} = \begin{cases} \alpha & \text{if } \boldsymbol{S}_{\boldsymbol{I}_{xy}} = 1 \\ 1 - \alpha & \text{otherwise.} \end{cases} \tag{A32}$$

where α represents the balance factor between the symmetry and non-symmetry pixels. We set $\alpha = 0.85$ for reflection symmetry detection and $\alpha = 0.95$ for rotation since reflection axes contain more positive pixels than rotation centers. The focusing parameter λ in Eq. (16) is set to 2.0.

Other training details. We trained CLIPSym for 500 epochs with a batch size of 16 on a single NVIDIA A100 80GB GPU. A single epoch takes around 5 minutes and the total training takes approximately 40 hours. We conducted a hyperparameter search within $\{10^{-3}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}, 5 \times 10^{-6}, 10^{-6}\}$ and found 10^{-5} is the best initial learning rate for both reflection and rotation symmetry detection. The difference is that reflection detection uses an exponential decay scheduler with a decay rate of 0.1, while rotation detection uses a constant learning rate.

Table A4. Details of the best prompts structure for symmetry detection.

	t_1	t_2	t_3	t_4	t_5
$\texttt{obj}_1$	tundra	ski pole	alphabet	tennis racket	race track
$\texttt{obj}_2$	orangutan	pass	snowboard	bright	martini glass
$\texttt{obj}_3$	maze	take	cap	control	windshield
$\texttt{obj}_4$	antler	sibling	champagne	construction site	stone building
	t_6	t_7	t_8	t_9	t_{10}
$\texttt{obj}_1$	damage	breakfast	sedan	pot	meter
$\texttt{obj}_2$	mouth organ	liquor	ceramic	pasture	skater
$\texttt{obj}_3$	driver	ladle	roll	driftwood	tripod
$\texttt{obj}_4$	snout	motorboat	toilet seat	folding chair	monster
	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
$\texttt{obj}_1$	coaster	padlock	mansion	bike lane	polka dot
$\texttt{obj}_2$	ceremony	concrete	scale	brownie	out
$\texttt{obj}_3$	shower door	spaghetti	zombie	color	keyboard
$\texttt{obj}_4$	signature	bee	potato	wine bottle	autumn
	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
$\texttt{obj}_1$	food stand	mill	spinach	crack	package
$\texttt{obj}_2$	design	handbag	underwater	teal	coffee
$\texttt{obj}_3$	scone	wet	knot	level	side table
$\texttt{obj}_4$	camouflage	clothe	deliver	tape	coconut
	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}
$\texttt{obj}_1$	dart	urban	enclosure	pottery	buffet
$\texttt{obj}_2$	city street	sweatshirt	screen door	bookcase	gravel
$\texttt{obj}_3$	blackberry	bowl	turret	shelter	apple
$\texttt{obj}_4$	goalkeeper	bike lane	tricycle	squad	rearview mirror