Supplementary Material for *SuS-X*: Training-Free Name-Only Transfer of Vision-Language Models

Table of Contents

A. Dataset Details

We enumerate the validation and testing split sizes of all datasets in Tab. 1. We make two small modifications to the standard datasets as described in CoOp [16]: (1) We discard the "BACKGROUND Google" and "Faces easy classes" from the Caltech101 dataset, and (2) For the UCF101 dataset, we consider the middle frame of each video as our image sample.

Table 1: Dataset details for the 19 datasets used in this study.

Dataset	Classes	Val	Test
UCF-101	101	1898	3783
CIFAR-10	10	10000	10000
CIFAR-100	100	10000	10000
Caltech101	100	1649	2465
Caltech256	257	6027	9076
ImageNet	1000	50000	50000
SUN397	397	3970	19850
FGVCAircraft	100	3333	3333
Birdsnap	500	7774	11747
StanfordCars	196	1635	8041
CUB	200	1194	5794
Flowers102	102	1633	2463
Food101	101	20200	30300
OxfordPets	37	736	3669
DTD	47	1128	1692
EuroSAT	10	5400	8100
ImageNet-Sketch	1000	50889	50889
ImageNet-R	200	30000	30000
Country211	211	10550	21100

B. Details about Support Set Curation Strategies

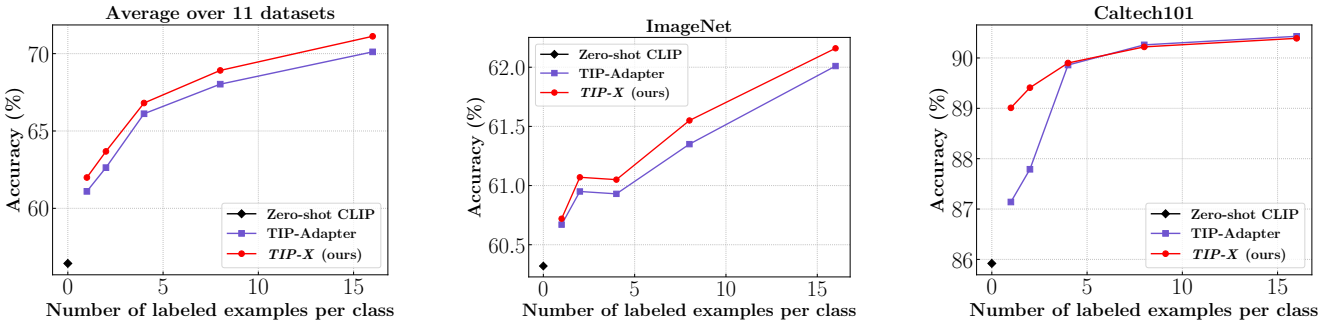
We include further technical details about our two *support set* curation strategies—Stable Diffusion Generation and LAION-5B Retrieval.

Stable Diffusion Generation. For all our experiments with the Stable Diffusion model, we use the stable-diffusion-v1-4 checkpoint with a 9.5 guidance scale [8], 85 diffusion steps and 512×512 output resolution. We then downscale these images to CLIP's input resolution of 224×224.

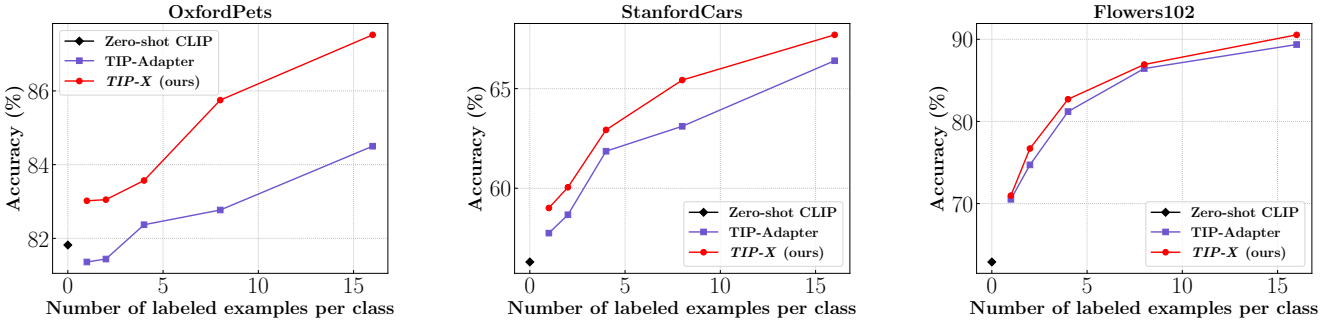
LAION-5B Retrieval. For all our experiments, we rank all images in the LAION-5B corpus based on their image-text similarity with the given class textual prompt. We use the LAION-5B pre-constructed index that leverages the CLIP-ViT-L/14 model. Finally, since the images might be of varying resolutions, we pre-process them to CLIP's input resolution of 224×224.

C. Few-shot learning with *TIP-X*

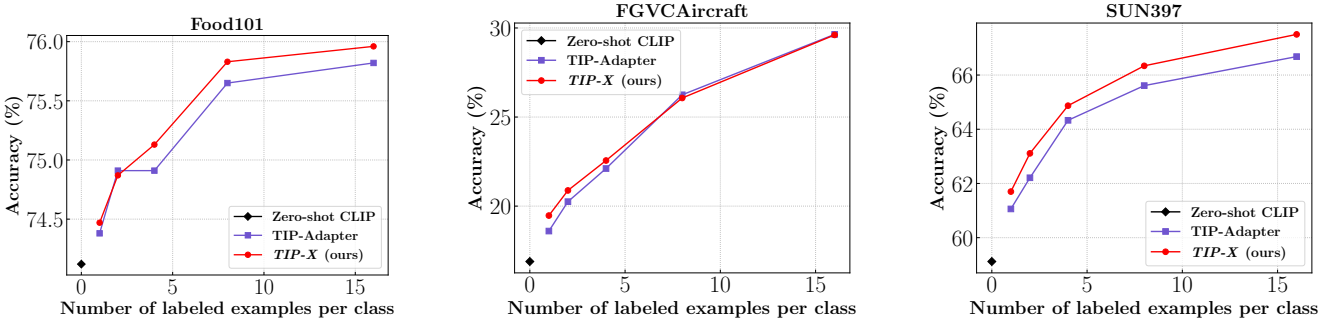
In Sec 4.3 of the paper, we adapt *TIP-X* to the few-shot training-free adaptation regime, and compare with the SoTA model TIP-Adapter. We now show the extended results on all 11 datasets in Fig. 1. On average, we outperform TIP-Adapter by 0.91% across all shots.



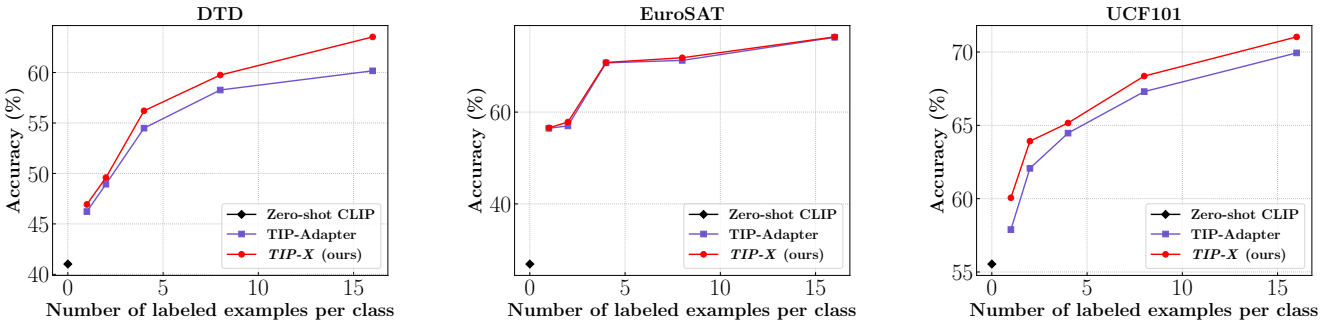
(a) Average (b) ImageNet (c) Caltech101



(d) OxfordPets (e) StanfordCars (f) Flowers102



(g) Food101 (h) FGVCAircraft (i) SUN397



(j) DTD (k) EuroSAT (l) UCF101

Figure 1: **Results for the training-free few-shot regime across 11 datasets.**

D. Details about Support Set Sizes

For our main results in Sec 4.1 of the paper, we use a fixed number of *support set* samples per dataset. In Tab. 2, we enumerate the number of *support set* samples used per dataset. As shown in Sec 4.4, the *support set* size can impact performance significantly—the nature of these impacts is dataset-specific.

Table 2: Support Set Sizes

Dataset	**Support Set Size**
UCF-101	5858
CIFAR-10	50
CIFAR-100	4700
Caltech101	101
Caltech256	3084
ImageNet	36000
SUN397	397
FGVCAircraft	7900
Birdsnap	39000
StanfordCars	980
CUB	400
Flowers102	3162
Food101	3434
OxfordPets	2627
DTD	188
EuroSAT	150
ImageNet-Sketch	42000
ImageNet-R	10200
Country211	844

E. Details about Baselines

For our main zero-shot/name-only training-free CLIP-based experiments, we use six main baselines—Zero-shot CLIP [12], CALIP [5], CLIP+DN [18], VisDesc [10], CuPL [11] and CuPL+e.

Zero-shot CLIP. For Zero-shot CLIP, we directly use the model weights released by OpenAI and the official inference scripts for reproducing results on different datasets[1]. For benchmarking all our results, we use the 7-prompt ensemble set used by TIP-Adapter [15] for all datasets. The 7 prompt templates in the ensemble are: "itap of a `<class>`.", "a origami `<class>`.", "a bad photo of the `<class>`.", "a photo of the large `<class>`.", "a `<class>` in a video game.", "art of the `<class>`.", and "a photo of the small `<class>`.".

CALIP details. Due to the unavailability of publicly released code at the time of writing this paper, we re-implement the CALIP baseline, following the description in [5]. We provide access to our re-implementation as part of our released codebase.

CLIP+DN details. For CLIP+DN, we use the official code[2] released by the authors on all datasets. As specified in the paper, we (i) use 100 random unlabeled validation samples for the mean estimation for DN, and (ii) report the average accuracy across 5 different random seeds.

VisDesc details. For VisDesc, we use the official code[3] released by the authors on all datasets. We use their default prompt settings for generating the GPT-3 descriptors.

CuPL details. For CuPL, we use the official code[4] released by the authors on all datasets. The list of pre-prompts used as inputs to GPT-3 for different datasets are listed in Tab. 3 and Tab. 4.

CuPL+e details. For CuPL+e, we simply concatenate the 7-prompt ensemble embeddings of each class with the custom GPT-3 generated CuPL embeddings of that particular class. We then average all the embeddings within a class to generate the textual embedding for that class. Then, we proceed as standard to construct the classifier weight matrix by stacking all class text embeddings.

E.1. Transfer to other VLMs

We can transfer all the aforementioned baselines to different VLMs by simply swapping out CLIP's frozen image and text encoders with those of TCL [14] and BLIP [9]. For the TCL[5] experiments, we use the standard ViT-B/16 base model that is fine-tuned for retrieval on MS-COCO, released by the authors here. For the BLIP[6] experiments, we use the standard ViT-B/16 base model fine-tuned for retrieval on MS-COCO, released by the authors here.

[1] https://github.com/openai/CLIP
[2] https://github.com/fengyuli2002/distribution-normalization
[3] https://github.com/sachit-menon/classify_by_description_release
[4] https://github.com/sarahpratt/CuPL
[5] https://github.com/uta-smile/TCL
[6] https://github.com/salesforce/BLIP

Table 3: CuPL hand-written prompts (1/2)

Dataset	**GPT-3 prompts**
UCF101	"What does a person doing {} look like" "Describe the process of {}" "How does a person {}"
CIFAR10	"Describe what a {} looks like" "How can you identify {}?" "What does {} look like?" "Describe an image from the internet of a {}" "A caption of an image of {}: "
CIFAR100	"Describe what a {} looks like" "How can you identify {}?" "What does {} look like?" "Describe an image from the internet of a {}" "A caption of an image of {}: "
Caltech101	"Describe what a {} looks like" "What does a {} look like" "Describe a photo of a {}"
Caltech256	"Describe what a {} looks like" "What does a {} look like" "Describe a photo of a {}"
ImageNet	"Describe what a {} looks like" "How can you identify {}?" "What does {} look like?" "Describe an image from the internet of a {}" "A caption of an image of {}: "
SUN397	"Describe what a {} looks like" "How can you identify a {}?" "Describe a photo of a {}"
FGVCAircraft	"Describe a {} aircraft"
Birdsnap	"Describe what a {}, a species of bird, looks like" "What does a {} look like" "Visually describe a {}, a type of bird" "A caption of an image of a {}, a type of bird" "Describe the appearance of a {}" "What are the prominent features to identify a {} bird"
StanfordCars	"How can you identify a {}" "Description of a {}, a type of car" "A caption of a photo of a {}:" "What are the primary characteristics of a {}?" "Description of the exterior of a {}" "What are the identifying characteristics of a {}, a type of car?" "Describe an image from the internet of a {}" "What does a {} look like?" "Describe what a {}, a type of car, looks like"

Table 4: CuPL hand-written prompts (2/2)

Dataset	GPT-3 prompts
CUB	"Describe what a {}, a species of bird, looks like" "What does a {} look like" "Visually describe a {}, a type of bird" "A caption of an image of a {}, a type of bird" "Describe the appearance of a {}" "What are the prominent features to identify a {} bird"
Flowers102	"What does a {} flower look like" "Describe the appearance of a {}" "A caption of an image of {}" "Visually describe a {}, a type of flower"
Food101	"Describe what a {} looks like" "Visually describe a {}" "How can you tell that the food in this photo is a {}?"
OxfordPets	"Describe what a {} pet looks like" "Visually describe a {}, a type of pet"
DTD	"What does a {} material look like?" "What does a {} surface look like?" "What does a {} texture look like?" "What does a {} object look like?" "What does a {} thing look like?" "What does a {} pattern look like?"
EuroSAT	"Describe an aerial satellite view of {}" "How does a satellite photo of a {} look like" "Visually describe a centered satellite view of a {}"
ImageNet-Sketch	"Describe how a black and white sketch of a {} looks like" "A black and white sketch of a {}" "Describe a black and white sketch from the internet of a {}"
ImageNet-R	"An art drawing of a {}" "Artwork showing a {}" "A cartoon a {}" "An origami of a {}" "A deviant art photo depicting a {}" "An embroidery of a " "A graffiti art showing a {}" "A painting of a {}" "A sculpture of a {}" "A black and white sketch of {}" "A toy of a {}" "A videogame of a {}"
Country211	"Visually describe what {} looks like" "What does the landscape of {} look like" "Describe a photo taken in {}" "How does a typical photo taken in {} look like"

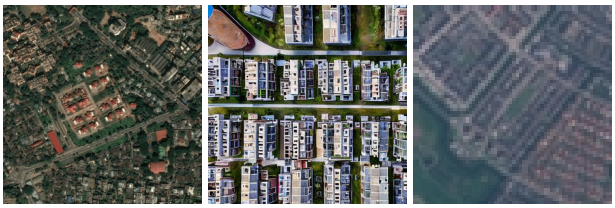
F. More *SuS* visualisations

In Fig. 2, we provide further *support set* samples across different datasets curated using both *SuS-LC* and *SuS-SD* methods.

(a) Birdsnap, Acadian Flycatcher

(b) Caltech101, Soccer Ball

(c) DTD, Chequered



(d) EuroSAT, Residential

(e) FGVCAircraft, A320

(f) Food101, Breakfast Burrito

(g) Flowers102, Water Lily

(h) OxfordPets, Persian Cat

(i) StanfordCars, Rolls Royce Ghost

(j) UCF101, Cricket Shot

Figure 2: **Support samples from the generated *SuS-SD*, retrieved *SuS-LC* and true training distribution for different datasets.** For each subfigure, the ordering of figures is—*SuS-LC*, *SuS-SD*, *Train*. We label each figure with its source dataset and class name.

G. Hyperparameter Settings

We provide the hyperparameter settings for obtaining our main results from Tab 2 in the main paper in Tab. 5. For our hyperparameters, we conduct a search over $[0.1, 50]$ for α, $[1, 50]$ for β and $[0.1, 30]$ for γ. In the main paper, we have a hyperparameter sensitivity test which ensures that the variance in accuracy values is not too large as we vary our hyperparameters.

Table 5: Hyperparameter settings for the 19 datasets.

Dataset	α	β	γ
UCF-101	0.10	8.59	0.10
CIFAR-10	5.09	5.41	0.10
CIFAR-100	0.10	1.49	0.10
Caltech101	0.10	1.27	0.10
Caltech256	0.10	12.76	0.10
ImageNet	10.08	39.46	0.10
SUN397	2.60	8.35	0.10
FGVCAircraft	2.60	24.52	0.69
Birdsnap	48.53	22.55	0.69
StanfordCars	0.10	1.58	0.10
CUB	0.10	8.84	0.10
Flowers102	0.10	2.72	0.10
Food101	17.56	49.02	0.10
OxfordPets	10.08	41.91	1.29
DTD	5.09	23.79	0.70
EuroSAT	2.60	1.00	0.10
ImageNet-Sketch	30.04	38.48	0.69
ImageNet-R	2.60	30.65	0.70
Country211	12.57	22.31	0.10

Results without tuning. We also report the results on all 19 datasets without tuning our hyperparameters in Tab. 6. For this, we fix the hyperparameters to be α=0.1, β=1.0, γ=0.1. Even without hyperparameter tuning, we see large gains over Zero-shot CLIP.

Table 6: Zero-shot/name-only results with fixed hyperparameters (no hyperparameter tuning)

	UCF101	CIFAR-10	CIFAR-100	Caltech101	Caltech256	ImageNet	SUN397	FGVCAircraft	Birdsnap	StanfordCars	CUB	Flowers102	Food101	OxfordPets	DTD	EuroSAT	ImageNet-Sketch	ImageNet-R	Country211	Average (11 subset)	Average (19 datasets)
ZS-CLIP	55.56	73.10	40.58	85.92	78.98	60.31	59.11	16.71	30.56	56.33	41.31	62.89	74.11	81.82	41.01	26.83	35.42	59.34	13.42	56.41	52.27
SuS-X-SD-P	61.41	74.68	43.45	89.57	80.46	61.64	62.96	18.84	36.20	57.19	48.90	66.18	77.45	85.17	48.76	37.11	36.05	61.69	14.26	60.57	55.89
SuS-X-SD-C	61.51	74.65	43.53	89.53	80.50	61.65	62.95	19.11	36.36	57.18	48.84	66.26	77.53	85.17	48.35	37.27	35.88	61.69	14.25	60.59	55.91
SuS-X-LC-P	61.49	74.62	44.30	89.57	80.56	61.80	63.02	20.04	36.75	57.19	48.81	66.87	77.36	85.31	47.87	37.49	36.25	61.62	14.20	60.73	56.01
SuS-X-LC-C	60.51	74.61	44.07	89.49	80.59	61.53	62.94	19.23	36.25	57.05	49.02	66.83	77.35	82.27	47.04	36.78	35.76	60.91	14.21	60.09	55.60

Analysis of hyperparameters. From Tab. 5, we note that for some datasets, the weight for the inter-modal distance term γ is dominated by the weight for the intra-modal distance term α. We analyse this in depth, and show that despite this disparity, using inter-modal distances still brings gains. Tab. 7 reports results on these datasets (for which $\alpha >> \gamma$) using their optimal

hyperparameters (α>γ), fixed hyperparameters (α=γ=0.1), and removed inter-modal contributions (γ=0). Apart from the unhighlighted rows, it is always beneficial to use small inter-modal distance contributions over neglecting them (see green rows). Hence, we conclude that both these terms are important for bringing the large performance gains of our model.

Table 7: **Analysis of α and γ values.**

Dataset	**Optimal** α>γ	**Fixed&Equal** α=γ=0.1	**Inter-modal only** α=optimal, γ=0
ImageNet	61.89	61.80	61.30
ImageNet-Sketch	37.83	36.25	36.10
ImageNet-R	62.10	61.62	61.30
OxfordPets	86.59	85.31	85.00
Birdsnap	38.50	36.75	37.70
Food101	77.62	77.53	77.55

H. Discussion on *SuS* vs CuPL/VisDesc

As discussed in the main paper, CuPL and VisDesc are two *name-only* transfer methods that leverage a large pre-trained language model (GPT-3) to enhance the textual prompts used for zero-shot classification. On the other hand, our *SuS* construction strategies endow the zero-shot model with rich visual information to discriminate between different categories.

We note that text alone cannot model the rich information in the world [2, 1]. Consider a task of classifying between two bird species—"Florida Scrub Jay" and "Blue Jay". The difference is all in the subtle visual details—blue jays have a crest and distinct black markings on their necks. This level of rich visual information is hard to extract from textual descriptions of class names. Hence, the main advantage of *SuS* is in imparting this expressive visual information for discriminating between fine-grained categories. We verify this empirically in Fig. 3 depicting large gains over CuPL+e (our best performing baseline) in fine-grained datasets like Birdsnap, Flowers102, OxfordPets etc (Full results in Tab. 11 below.).

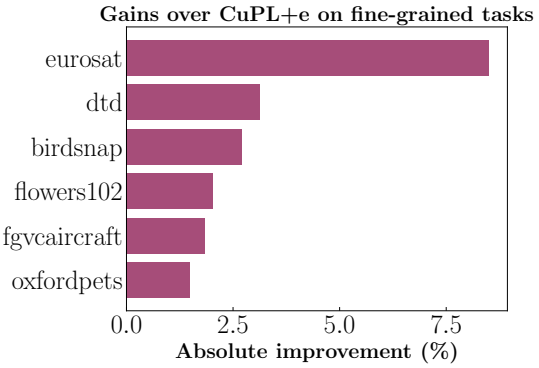


Figure 3: Improvement for fine-grained tasks.

I. Compute Cost Comparison

We compare the computational requirements of our ***SuS-X*** and the baselines in Tab. 8—for each method, we measure the time and memory requirements for one ImageNet class *i.e.* on 50 test images. For CuPL, VisDesc and ***SuS-X***, we measure the construction time required for curating the enhanced textual prompts and *support sets*. Note that in practical applications, it is typical to cache the curated *support sets*/prompts for each class, thereby amortising costs across queries. We note that our ***SuS-X*** models offer the most competitive performance-efficiency tradeoff when comparing the compute requirements and accuracy values.

Table 8: Compute requirements.

Method	Construction Time	Inference Time	GPU Memory	ImageNet Accuracy
Zero-shot	–	10.22ms	2.2GB	60.32
CALIP	–	121.26ms	24GB	60.57
CLIP+DN	–	10.22ms	2.2GB	60.16
VisDesc	∼3s	10.22ms	2.2GB	59.68
CuPL+e	∼3s	10.22ms	2.2GB	61.64
SuS-X-SD	∼60s	10.50ms	3.2GB	61.84
SuS-X-LC	∼2s	10.50ms	3.2GB	61.89

*Tested on single Nvidia A100-80GB GPU with one ImageNet class (50 test images).

J. Diversity of *CuPL* and *Photo* prompting strategies

In this section, we describe in detail the computation of the diversity metric used in Sec 4.4 of the paper.

We assume access to a support set S of size NC, where there are C classes and N support samples per class. We denote the support subset of a given class i as $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,N}\}$, where $s_{i,j}$ denotes the j^{th} support sample for class i. Corresponding to these support subsets, we denote the features of S_i as F_i (using CLIP's image encoder):

$$F_{i,j} = \texttt{CLIPImageEncoder}(s_{i,j}), F_{i,j} \in \mathbb{R}^d, i \in [1, C], j \in [1, N]$$
$$F_i = \texttt{Concat}([F_{i,1}, F_{i,2}, \ldots, F_{i,N}]), F_i \in \mathbb{R}^{N \times d}$$

We now compute the mean pairwise cosine similarity between all support samples within a class *i.e.* for class i, we compute:

$$\texttt{PCS}_i = \frac{\sum_{j=1}^{N} \sum_{k=1}^{N} F_{i,j} F_{i,k}^T}{N^2}$$

The intuition is that if all the support samples within a class are similar to each other, then the support set is less diverse. Hence, a higher value of $\texttt{PCS}_i$ implies a lower diversity. We then compute the mean PCS over all classes as:

$$\texttt{MPCS} = \frac{\sum_{i=1}^{C} \texttt{PCS}_i}{C}$$

Finally, we define diversity to be:

$$\texttt{Diversity} = 1 - \texttt{MPCS}$$

K. Further Analyses and Discussions

We conduct some further ablation studies to analyse our method with more rigour. Due to lack of space in the main paper, we include these ablations here, however these are vital analyses which delineate important properties of our method.

K.1. Contribution of intra-model and inter-modal distances

In Sec 3.2 of the paper, we describe our *TIP-X* method that utilises image-text distances as a bridge for computing image-image intra-modal similarities. We refer to the main equation for computing *TIP-X* logits again, highlighting the importance of each term:

$$\texttt{TXL} = \underbrace{fW^T}_{\text{1. zero-shot component}} + \underbrace{\alpha AL}_{\text{2. intra-modal distance component}} + \underbrace{\gamma\psi(-M)L}_{\text{3. inter-modal distance component}}$$

Zero-shot CLIP utilises only the zero-shot term (**1**) above. TIP-Adapter utilises the zero-shot and intra-modal distance terms (**1+2**). Our method uses all three terms (**1+2+3**). We further ablate this design choice to break down the gains brought forth from each individual term. In Tab. 9, we show the performance gains from each of these terms with our best performing ***SuS-X-LC*** model across 19 datasets. We observe large gains from inter-modal and intra-modal distances independently over just using the zero-shot term. Further, both these distances provide complementary information to each other, and hence can be productively combined leading to the best results.

Table 9: Contribution of intra-modal and inter-modal distances.

Dist. terms used	**1** (Zero-shot)	**1+3** (Inter-modal)	**1+2** (Intra-modal)	**1+2+3** (Both)
Average Acc.	52.27	56.30	56.56	56.87
Gain	0	+4.03	+4.29	+4.60

K.2. Comparing *name-only SuS-X* to few-shot methods

In Sec 4.1 of the main paper, we showcased state-of-the-art results with our ***SuS-X*** model in the *name-only* setting. Recollect that in this setting, we use no images from the true target distribution. Here, we evaluate how well our ***SuS-X*** model fares against methods that use image samples from the true target distribution. We compare our best performing ***SuS-X-LC*** method (uses no images from target distribution) with 16-shot TIP-Adapter and 16-shot *TIP-X* (both using 16 labelled images per class). From Tab. 10, we see that ***SuS-X-LC*** is competitive (in green) against these few-shot adaptation methods, despite using no target task images. There are however cases where ***SuS-X-LC*** severely underperforms the few-shot methods—this is due to the domain gap between the *SuS* images and the true labelled images (refer Sec 4.4 of the paper).

Table 10: *SuS-X* against few-shot labelled methods.

Dataset	**Zero-shot**	***SuS-X-LC*** (name-only, ours)	**TIP-Adapter** (few-shot)	***TIP-X*** (few-shot, ours)
ImageNet	60.31	61.89	62.01	62.16
Food101	74.11	77.62	75.82	75.96
OxfordPets	81.82	86.59	84.50	87.52
Caltech101	85.92	89.65	90.43	90.39
Flowers102	62.89	67.97	89.36	90.54
FGVCAircraft	16.71	21.09	29.64	29.61

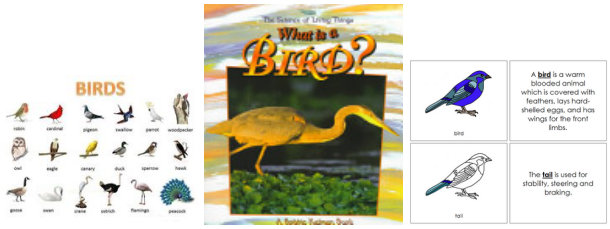
K.3. Intuitions for best performing configurations

From Tab 5 of the main paper, we note that our best *name-only* results are achieved with the LC-Photo and SD-CuPL *SuS* construction strategies. A natural question arises: "*Why do the two SuS construction methods require different prompting strategies for achieving their best results?*". We attempt to answer this question via careful inspection of the *support sets* curated from these strategies. For this case study, we inspect the *support sets* for the CIFAR-10 dataset.

(a) *SuS-LC*, *Photo*, Airplane

(b) *SuS-LC*, *CuPL*, Airplane

(c) *SuS-LC*, *Photo*, Bird



(d) *SuS-LC*, *CuPL*, Bird

(e) *SuS-SD*, *Photo*, Airplane

(f) *SuS-SD*, *CuPL*, Airplane

(g) *SuS-SD*, *Photo*, Bird

(h) *SuS-SD*, *CuPL*, Bird

Figure 4: **Uncovering the intuitions for different prompting configurations.** We showcase some support samples using different prompting configurations for two CIFAR-10 classes—*airplane* and *bird*. The key takeaways upon inspecting these samples are enumerated below.

From Fig. 4, we can draw two key takeaways regarding the best prompting strategies for the two *SuS* curation methods:

1. **LAION-5B retrieval.** The *support sets* constructed with *CuPL* prompts are largely divergent from the "true" distribution of natural semantic images of the target concepts/classes. This can be noted from the right panels of the first two rows in Fig. 4—this disparity in the retrieved *support set* images leads to a large domain gap to the target distribution, hence resulting in poorer performance than the *Photo* prompting strategy. Further, since the LAION-5B *support sets* consist of natural images *i.e*. images available on the web, the LAION-5B *Photo support set* images are closer to the true target distribution of images.

2. **Stable Diffusion Generation.** The *support sets* generated using Stable Diffusion represent a synthetic data distribution *i.e*. there is an innate distribution shift from the target distribution images owing to the target datasets (mostly) consisting of natural images. Hence, the Stable Diffusion *support sets* are inherently at a disadvantage compared to the LAION-5B *support sets*. However, within the constructed Stable Diffusion *support sets*, the *CuPL* prompting strategy mildly wins over the *Photo* strategy since it helps generate a more diverse set of images (consisting of more expansive lighting conditions, background scenes etc.)—this diversity helps reduce the domain gap to the target dataset to a small extent. This phenomenon of added diversity in synthetic datasets aiding downstream performance has also been noted in previous works [7].

L. Extended Results on all Datasets

In Tab. 11, we report the accuracies obtained on each of the 19 individual datasets for all our baselines, and our ***SuS-X*** model variants with CLIP as the VLM. We also report the average accuracy obtained on the 11 dataset subset used in previous CLIP adaptation works [15, 4, 16]. In Tab. 12, we report all the results with the TCL model as the VLM, and in Tab. 13, we report the results with the BLIP model as the VLM.

Table 11: **Training-free zero-shot/name-only results across model configurations and datasets.** We report average results using both the 11 dataset subset used by previous works on few-shot adaptation [15, 4, 16] and the entire 19 dataset suite. For the CALIP baseline, we report numbers from the original paper (denoted with a subscript o) as well as our re-implementation (denoted with a subscript (r)). We refer to the Zero-shot CLIP model as ZS-CLIP and CuPL+ensemble baseline as CuPL+e. We use the CuPL+ensemble prompts for CLIP's text classifier in this experiment. For both variants of our models, we append P or C to the name to distinguish between *Photo* and *CuPL* prompt strategies. For instance, *SuS-X-LC-P* refers to the *SuS-X* model with LC curation using the *Photo* strategy. All models use the ResNet-50 visual backbone. The best results for each dataset are **bolded** and the second best are underlined. This table contains the full set of values used for generating Fig 4a and populating Tab 2 in the paper.

	UCF101	CIFAR-10	CIFAR-100	Caltech101	Caltech256	ImageNet	SUN397	FGVCAircraft	Birdsnap	StanfordCars	CUB	Flowers102	Food101	OxfordPets	DTD	EuroSAT	ImageNet-Sketch	ImageNet-R	Country211	Average (11 subset)	Average (19 datasets)
ZS-CLIP	55.56	73.10	40.58	85.92	78.98	60.31	59.11	16.71	30.56	56.33	41.31	62.89	74.11	81.82	41.01	26.83	35.42	59.34	13.42	56.41	52.27
CALIP$_o$	**61.72**	–	–	87.71	–	60.57	58.59	17.76	–	56.27	–	66.38	77.42	86.21	42.39	38.90	–	–	–	59.45	–
CALIP$_r$	55.61	73.15	40.62	86.20	79.08	60.31	59.10	16.71	30.68	56.32	41.40	63.01	74.13	81.84	41.01	26.96	36.10	59.32	13.45	56.47	52.37
CLIP+DN	55.60	74.49	43.73	87.25	79.24	60.16	59.11	17.43	31.23	56.55	43.03	63.32	74.64	81.92	41.21	28.31	35.95	60.37	13.76	56.86	53.02
CuPL	58.97	74.13	42.90	89.29	80.29	61.45	62.55	19.59	35.65	57.28	48.84	65.44	76.94	84.84	48.64	38.38	35.13	61.02	13.27	60.30	55.50
CuPL+e	61.45	74.67	43.35	89.41	80.57	61.64	62.99	19.26	35.80	57.23	48.77	65.93	77.52	85.09	47.45	37.06	35.85	61.17	**14.27**	60.45	55.76
VisDesc	58.47	73.22	39.69	88.11	79.94	59.68	59.84	16.26	35.65	54.76	48.31	65.37	76.80	82.39	41.96	37.60	33.78	57.16	12.42	58.30	53.76
SuS-X-SD-P	**61.72**	74.71	44.14	**89.65**	80.62	61.79	62.96	19.17	36.59	**57.37**	**49.12**	**67.97**	77.59	86.24	49.35	38.11	36.58	**62.10**	14.26	61.08	56.32
SuS-X-SD-C	61.54	74.69	**44.63**	89.53	**80.64**	61.84	62.95	19.47	37.14	57.27	**49.12**	67.72	77.58	85.34	**50.59**	**45.57**	36.30	61.76	**14.27**	**61.76**	56.73
SuS-X-LC-P	61.49	**74.95**	44.48	89.57	80.62	**61.89**	**63.01**	**21.09**	**38.50**	57.17	48.86	67.07	**77.62**	**86.59**	49.23	44.23	**37.83**	**62.10**	14.24	61.72	**56.87**
SuS-X-LC-C	61.43	74.76	44.12	89.61	80.63	61.79	62.94	20.34	37.07	57.06	48.86	67.60	77.58	85.22	49.47	37.16	36.45	61.39	14.26	60.93	56.20

Table 12: **Training-free zero-shot/name-only results across model configurations using the TCL [14] architecture.** For our *SuS-X* models, we only use the two best configurations from the previous CLIP experiment *i.e. SuS-X-SD* with *CuPL* strategy and *SuS-X-LC* with *Photo* strategy. This table contains the full set of values used for populating Tab 3 in the paper.

	UCF101	CIFAR-10	CIFAR-100	Caltech101	Caltech256	ImageNet	SUN397	FGVCAircraft	Birdsnap	StanfordCars	CUB	Flowers102	Food101	OxfordPets	DTD	EuroSAT	ImageNet-Sketch	ImageNet-R	Country211	Average (11 subset)	Average (19 datasets)
ZS-TCL	35.29	82.33	50.86	77.65	61.90	35.55	42.12	2.25	4.51	1.53	7.63	28.30	24.71	20.63	28.55	20.80	24.24	46.05	1.42	28.84	31.38
CuPL	41.23	81.75	52.63	**81.66**	65.91	41.60	49.35	3.48	6.83	2.11	**10.20**	26.10	23.62	22.15	42.84	26.30	25.67	53.61	**4.07**	32.77	34.79
CuPL+e	41.63	82.07	52.66	81.29	66.46	41.36	49.98	3.51	6.60	2.11	9.80	26.91	24.84	21.17	41.96	25.88	26.36	53.36	3.68	34.82	32.79
VisDesc	42.53	82.30	51.89	77.00	66.51	40.40	**51.18**	3.21	5.69	**2.91**	8.96	25.13	27.16	24.58	34.28	21.27	27.05	49.26	3.57	31.77	33.94
SuS-X-SD-C	47.66	82.92	55.19	81.38	66.52	52.29	49.98	9.21	13.60	2.31	9.72	**30.98**	**48.87**	65.96	**48.17**	28.75	32.22	**58.95**	3.66	42.32	41.49
SuS-X-LC-P	**50.28**	**83.14**	**57.47**	81.38	**66.80**	**52.77**	49.97	**10.98**	**17.93**	2.57	9.77	30.04	48.06	**69.96**	46.63	**36.90**	**36.28**	57.58	3.72	**43.59**	**42.75**

*We use the official TCL-base checkpoint from here for these results.

Table 13: **Training-free zero-shot/name-only results across model configurations using the BLIP [9] architecture.** For our *SuS-X* models, we only use the two best configurations from the previous CLIP experiment *i.e. SuS-X-SD* with *CuPL* strategy and *SuS-X-LC* with *Photo* strategy. This table contains the full set of values used for populating Tab 3 in the paper.

	UCF101	CIFAR-10	CIFAR-100	Caltech101	Caltech256	ImageNet	SUN397	FGVCAircraft	Birdsnap	StanfordCars	CUB	Flowers102	Food101	OxfordPets	DTD	EuroSAT	ImageNet-Sketch	ImageNet-R	Country211	Average (11 subset)	Average (19 datasets)
ZS-BLIP	50.49	86.68	61.72	92.13	82.17	50.59	54.22	5.40	10.21	54.71	14.95	40.15	54.21	59.04	44.68	44.10	43.69	70.93	5.84	49.97	48.73
CuPL	56.09	86.06	61.99	**92.41**	83.45	52.96	59.16	5.85	12.24	54.64	18.53	43.97	56.14	72.00	52.95	39.37	44.83	72.27	6.26	53.23	51.11
CuPL+e	55.61	86.33	62.16	92.29	83.59	53.07	59.38	6.27	12.18	54.89	18.63	43.72	57.10	71.73	53.30	41.48	45.34	72.40	6.42	53.53	51.36
VisDesc	53.42	86.78	60.47	92.04	81.53	50.94	55.85	6.30	11.69	54.64	16.65	42.71	58.50	69.22	47.45	42.25	43.30	68.62	6.01	52.12	49.91
SuS-X-SD-C	57.28	87.56	63.60	92.33	**83.66**	55.93	**59.46**	10.14	16.95	54.89	18.95	**44.38**	62.75	74.68	**56.15**	45.36	46.51	**73.85**	**6.45**	55.76	53.20
SuS-X-LC-P	**59.90**	**88.28**	**64.43**	92.29	83.61	**56.75**	59.39	**11.82**	**23.78**	**54.94**	**19.24**	43.97	**64.14**	**79.72**	55.91	**51.62**	**48.53**	73.42	6.44	**57.31**	**54.64**

*We use the official BLIP-base checkpoint from here for these results.

M. Results with different Visual Backbones

All our main results use the ResNet-50 [6] visual backbone for CLIP's image encoder. In Tab. 14, we compare the accuracies obtained on all 19 datasets using 2 different visual backbone model classes—ResNets [6] (ResNet-50, ResNet-101) and Vision Transformers [3] (ViT-B/32, ViT-B/16). We observe that the accuracy values monotonically improve as we increase the model capacity.

N. Results with different Text-to-Image Generation Models

We also experiment with different text-to-image generation models for *support set* generation to showcase the generalisability and robustness of our method's results. Tab. 15 depicts ***SuS-X-SD*** results by generating *support sets* using different text-to-image generation models. The results presented in the main paper all use the Stable-Diffusion-v1.4 model, but we also note similar gains across three other generative models.

Table 14: **Training-free name-only results across visual backbones.** For this experiment, we use the default versions of our ***SuS-X*** models: ***SuS-X-LC*** with *Photo* strategy and ***SuS-X-SD*** with *CuPL* strategy. This experiment uses the CuPL prompts for CLIP's text classifier. This table contains the raw data for generating **??** of the paper.

		UCF101	CIFAR-10	CIFAR-100	Caltech101	Caltech256	ImageNet	SUN397	FGVCAircraft	Birdsnap	StanfordCars	CUB	Flowers102	Food101	OxfordPets	DTD	EuroSAT	ImageNet-Sketch	ImageNet-R	Country211	Average (11 subset)	Average (19 datasets)
RN50	*SuS-X-LC*	59.98	74.79	44.22	89.29	80.29	61.66	62.70	21.87	38.56	56.92	48.90	66.91	77.21	86.35	50.06	43.99	37.25	61.97	13.21	61.54	56.64
	SuS-X-SD	59.48	74.21	44.33	89.25	80.27	61.65	62.58	19.92	37.00	57.14	49.10	67.32	77.02	85.09	51.00	47.69	37.25	61.73	13.30	61.65	56.59
RN101	*SuS-X-LC*	60.03	77.51	46.72	92.09	81.96	62.11	61.50	22.92	39.87	61.20	45.82	59.28	78.52	88.44	51.18	39.23	40.05	69.07	11.45	61.50	57.31
	SuS-X-SD	57.84	76.97	46.01	92.09	81.96	62.18	61.61	21.66	35.60	61.05	45.93	60.90	78.41	86.56	51.95	39.23	40.47	68.94	11.41	61.23	56.88
ViT-B/32	*SuS-X-LC*	63.49	89.32	65.25	93.18	84.73	64.73	65.49	23.01	40.77	61.19	53.03	68.01	80.31	87.95	52.25	53.91	43.10	70.55	14.91	64.87	61.85
	SuS-X-SD	63.20	88.39	64.84	93.18	84.73	64.71	65.47	21.66	38.97	61.12	53.52	68.17	80.24	86.81	51.89	53.91	43.27	70.42	14.91	64.58	61.55
ViT-B/16	*SuS-X-LC*	66.72	90.94	68.66	93.91	87.41	70.00	67.85	30.51	47.71	65.90	56.96	73.08	86.08	91.58	55.32	58.06	49.34	78.20	19.19	69.00	66.18
	SuS-X-SD	66.59	89.88	68.47	93.96	87.45	69.88	67.73	28.68	45.53	66.13	57.11	73.81	86.08	90.57	54.55	57.49	49.51	78.22	19.28	68.68	65.84

Table 15: ***SuS-X-SD* Results with additional T2I models.**

T2I Model	ImageNet	EuroSAT	DTD	OxfordPets	Average
ZS-CLIP (baseline)	60.31	26.83	41.01	81.82	52.49
StableDiffusion-1.4 (from main paper)	**61.84**	45.57	50.59	85.34	60.84 (+8.35%)
Kandinsky2.1	61.83	44.96	49.17	**85.47**	60.36 (+7.87%)
OpenJourney-4	61.81	45.00	**50.71**	85.17	60.67 (+8.18%)
Protogen-2.2	61.82	**48.67**	50.35	85.26	**61.52** (+9.03%)

O. Fine-tuning *SuS-X*

Despite our work's main focus being the training-free adaptation regime, we explore some preliminary results with fine-tuning ***SuS-X*** on a few datasets. We compare both the training-free and the fine-tuned variants of ***SuS-X*** with other CLIP adaptation methods that use full or partial (parameter-efficient fine-tuning) in Tab. 16. We note for some datasets, full/partial fine-tuning methods perform better than training-free ***SuS-X***. However, due to the domain gap between StableDiffusion/LAION-5B curated data and real test data, the gains are not large (confirming prior work [7, 13]). Further, we note that full fine-tuning and ***SuS-X*** are complementary, allowing a large boost in performance for ***SuS-X-F***. On the other hand, we emphasise that the goal of our work is to keep the approach *flexible* and *scalable*—one can apply ***SuS-X*** to an arbitrary number of rare categories *without training*. This training-free approach can particularly benefit when the categories of interest vary frequently, *rendering repetitive fine-tuning inefficient*. Moreover, fine-tuning forces the model to fit a very specific task distribution, enforcing forgetting of the model's pre-trained performance on a wide array of tasks. Since ***SuS-X*** only requires target task class names and does not fine-tune the model, we can cache the task-specific *support sets* a-priori and switch them dynamically based on the task at hand, without causing catastrophic forgetting of CLIP's pre-trained knowledge.

Table 16: **Fine-tuning methods vs *SuS-X*.**

Method	ZS-CLIP (No adaptation)	FT-CLIP (Full fine-tuning)	CoOp [17] (PromptTuning)	CLIP-Adapter [4] (Adapters)	***SuS-X*** **(Ours)**	***SuS-X-F*** **(Ours)**
ImageNet	60.31	60.35	60.96	61.61	<u>61.89</u>	**63.22**
EuroSAT	26.83	55.37	52.12	<u>57.00</u>	44.23	**59.22**
DTD	41.01	<u>50.35</u>	45.66	49.29	49.23	**52.30**
OxfordPets	81.82	84.51	85.99	85.06	<u>86.59</u>	**87.77**

References

[1] Jacob Browning and Yann Lecun. Ai and the limits of language, 2022. 10

[2] Nigel H Collier, Fangyu Liu, and Ehsan Shareghi. On reality and the limits of language data. *arXiv preprint arXiv:2208.11981*, 2022. 10

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 15

[4] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 14, 16

[5] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. Calip: Zero-shot enhancement of clip with parameter-free attention. *arXiv preprint arXiv:2209.14169*, 2022. 5

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 15

[7] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 13, 16

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[9] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 5, 15

[10] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022. 5

[11] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *arXiv preprint arXiv:2209.03320*, 2022. 5

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5

[13] Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR 2023–IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 16

[14] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15671–15680, 2022. 5, 15

[15] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. *arXiv preprint arXiv:2207.09519*, 2022. 5, 14

[16] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 2, 14

[17] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 16

[18] Yifei Zhou, Juntao Ren, Fengyu Li, Ramin Zabih, and Ser-Nam Lim. Distribution normalization: An "effortless" test-time augmentation for contrastively learned visual-language models. *arXiv preprint arXiv:2302.11084*, 2023. 5