AnomalyVFM – Transforming Vision Foundation Models into Zero-Shot Anomaly Detectors

Supplementary Material

In this Appendix, we provide extensive additional details and supporting information that extend beyond the scope of the main manuscript. The Appendix is organised as follows:

- **Limitations** in Section A.
- **Discussion about the Dataset Generation Phase** in Section B.
- **Results of competing methods when trained on the synthetic dataset and a discussion about them** in Section C.
- **Extended synthetic dataset details** in Section D.
- **Extended ablation studies** in Section E.
- **Additional qualitative results** in Section F.
- **Data generation data** in Section G.

A. Limitations

The main limitation currently is the time required to generate the synthetic dataset, which takes approximately one day on an A100 GPU, whereas model training requires only about two hours. While this represents a lot of time, it is a one-time investment, and the same dataset can be used for every VFM. With the improvements to the generation speed of current image generation models, we expect this time to drop even further. In Section E, we also conducted additional experiments, demonstrating that good performance can be achieved with fewer than 10,000 images, meaning the generation phase can be shorter if needed.

Additionally, while AnomalyVFM performs well on medical datasets, its performance could be further improved. In our preliminary attempts, the pretrained image generation models [3, 6, 7] failed to output realistic medical images suitable for zero-shot anomaly detection training. While this was not needed for industrial anomaly detection, fine-tuning the image generator on an auxiliary medical imaging dataset may enable the image-generation model to output data of suitable quality.

B. Discussion about dataset generation phase

While our synthetic dataset generation works well, it could be further improved. More specifically, the anomaly mask estimation and image filtering could be further improved. Although the dataset filtering is quite robust, some images without anomalies still pass through. Some examples of this can be seen in Figure 1. A trained AnomalyVFM could be used to further filter the data and improve the data quality even further. On top of that, the amount and the content of `[Object]` tags could be improved. Based on the experiment in Section E, we hypothesise that this would improve the performance even further. We have, however, left this for future work.

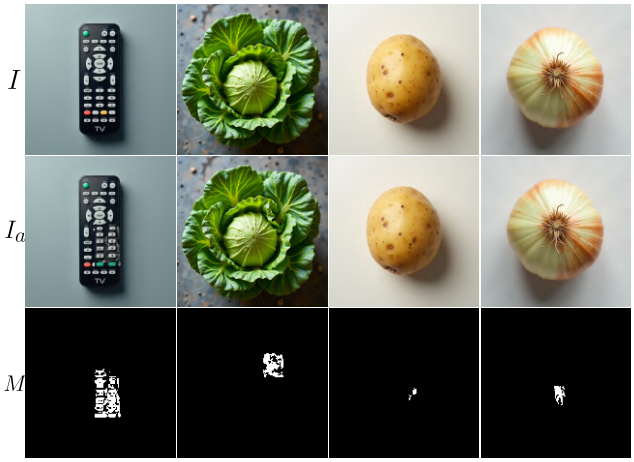

Figure 1. Failure Cases in Image Generation Process

Table 1. Comparison of performance of competing methods when trained on the proposed synthetic dataset versus when using the default datasets. SD stands for Synthetic Dataset

Method	SD	Image-level		Pixel-level	
		AUROC	F1-Max	AUROC	F1-Max
AACLIP [4]		86.1	81.4	95.0	45.1
	✓	85.6 ↓ 0.5	81.1 ↓ 0.3	93.5 ↓ 1.5	44.2 ↓ 0.9
AnomalyCLIP [8]		87.6	83.6	95.1	46.0
	✓	87.5 ↓ 0.1	82.5 ↓ 1.1	95.3 ↑ 0.3	45.8 ↓ 0.2
FAPrompt [9]		88.1	84.1	95.9	46.6
	✓	88.5 ↑ 0.4	84.4 ↑ 0.3	96.2 ↑ 0.3	45.9 ↓ 0.7
AdaCLIP [2]		89.6	87.5	95.0	51.0
	✓	87.1 ↓ 2.5	84.9 ↓ 2.6	92.9 ↓ 2.1	47.9 ↓ 3.1
Bayes-PFL [5]		90.8	85.1	96.0	43.9
	✓	91.2 ↑ 0.4	85.6 ↑ 0.5	96.1 ↑ 0.1	43.8 ↓ 0.1

To ensure that **no data leakage** occurred during the generation phase, we manually reviewed the `[Object]` tags and excluded any tags that were included in the evaluation test sets. We have left `[Anomaly]` and `[Texture]` as they were generated, as these represent more general concepts.

C. Training Competing methods with the proposed synthetic dataset

To demonstrate that the diversity of the datasets is not problematic for VLM-based methods, we retrained them using the proposed synthetic dataset. The results can be seen in

Table 2. Dataset Statistics for the generated dataset

Dataset Statistic	Value
No. of images	10,000
No. of different objects	100
No. of different backgrounds	50
No. of different anomalies	204
No. of object background combinations	4,596
Avg. Anomalous Area	2.52%
Min. Anomalous Area	0.28%
Max. Anomalous Area	11.24%

Table 1. While it does help for some methods, it does not significantly alter the results. This indicates that VLM methods do not suffer from the same problem of inadequate data diversity as VFMs.

D. Synthetic Dataset Details

Here, we provide details about the synthetic dataset generated for training our model. High-level statistics can be seen in Table 2. The generated dataset contains all of the possible objects and backgrounds. Additionally, it contains 204 different anomalies, significantly more than current datasets (e.g. MVTec AD [1] contains 73 different anomaly types). The generated anomalies are, in general, relatively small (on average, they account for 2.52% of the image). In contrast, in MVTec AD, they occupy 4.39% of the image. A more detailed visualisation of the anomaly area distribution is depicted in Figure 2.

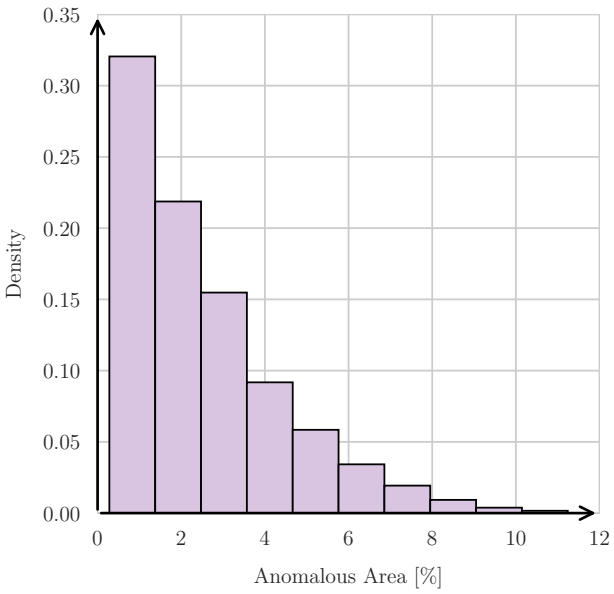

Figure 2. Anomalous Area Distribution in the generated synthetic dataset.

E. Additional Ablation Studies

In this section, we present additional experiments that verify the design choices in AnomalyVFM. Most of the results are presented in Table 3

Filtering Threshold To verify the impact of the threshold T used during dataset filtering, we re-filtered the dataset using various values of T. On top of the performance metrics, we also measured the rejection rate (i.e., the percentage of images discarded). The results and the rejection rate can be seen in Figure 3. The results show that the image-level AUROC is quite robust to the set threshold, while the pixel-level AUROC is more reliant on a correct choice of a threshold. At the default setting, the rejection rate is approximately 30%, showcasing that prompt adherence is far from a solved problem in generative models.

Number of `[Object]` tags To verify the importance of having a diverse dataset, we varied the number of `[Object]` tags during the synthetic data generation phase. The results can be seen in Figure 4. The results consistently rise with the number of `[Object]` tags. The performance with 20 `[Object]` tags is similar to the performance when AnomalyVFM is trained on MVTec AD [1], which has 15 different objects inside the dataset. We have not gone above 100 tags, as that is the list we initially generated with an LLM. In the future, we will increase this to see if the performance can be improved even further.

Number of images During all of our experiments, we used 10,000 generated images. To verify the importance of this, we have tried several different quantities: 100, 500, 1,000, and 10,000. The results are depicted in Figure 5. The performance increases steadily with each increment. We hypothesise that further scaling could improve performance even further. We have not done so to maintain a training set size similar to that of the related methods.

LoRA Rank To verify the robustness of the proposed method towards the rank of the LoRA adapters, we varied this parameter. More specifically, we decreased the rank to 32 and then increased it to 128. Decreasing it leads to a decrease of 0.3 p. p. in image-level AUROC and 0.3 p. p. in pixel-level AUROC. Increasing the LoRA rank leads to no differences in image-level metrics, while the pixel-level AUROC decreases for 0.3 p. p. This shows the robustness of the proposed method to this parameter.

LoRA Positions In the implementation, LoRA is added to query, value and projection layers inside the attention mechanism. This was done based on the insights from the open-source community on how to efficiently adapt image generation models. To verify the importance of this choice, we performed experiments with more layouts. All of the layouts keep a similar performance, showcasing robustness to this choice. The largest dip in performance is observed when LoRA adaptors are added to all linear layers. We assume this

Table 3. Additional ablations of the anomaly detection method components.

Group	Condition	Image-level		Pixel-level	
		AUROC	F_1-Max	AUROC	F_1-Max
Module Ablation	ViT-L → ViT-B	-1.8	-1.9	-1.2	-2.5
	ViT-L → ViT-H	-0.6	-0.4	-0.8	-2.3
	LoRA Rank 64 → 32	-0.3	0.0	-0.3	+0.4
	LoRA Rank 64 → 128	0.0	+0.1	-0.3	-0.2
	LoRA Positions: QKV and Proj	-0.1	+0.2	-0.2	-0.2
	LoRA Positions: All Norm Layers	-0.1	-0.4	-0.4	-1.1
	LoRA Positions: All Linear Layers	-0.3	-1.3	-0.1	-0.5
AnomalyVFM	LoRA Positions: QV and Proj	94.1	87.6	96.9	44.3

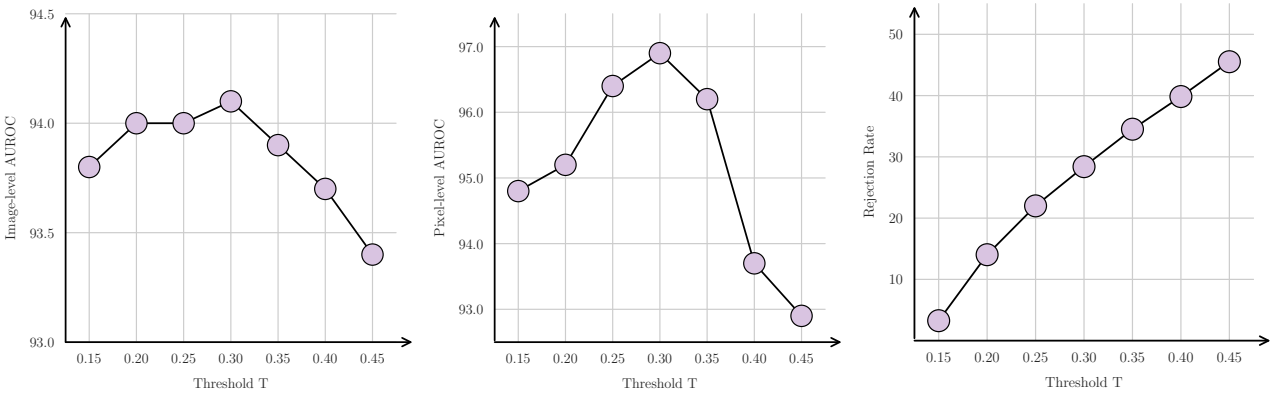

Figure 3. Model performance and rejection rate in relation to filtering threshold T.

is the case as the model cannot pass the information globally but rather only locally.

Model Size RADIO has multiple model sizes. To verify the importance of this parameter, we exchanged it with a smaller (ViT-B) and larger (ViT-H) model. Using a smaller model leads to a decrease of 1.8 p. p. in image-level AUROC and 1.2 p. p. in pixel-level AUROC. A larger model leads to a decrease of 0.6 p. p. in image-level AUROC and 1.2 p. p. in pixel-level AUROC. This shows that ViT-L is the optimal choice. We also hypothesise that increasing the number of `[Object]` tags and the total number of images would make ViT-H more optimal.

F. Additional Qualitative Examples

In this section, we add additional qualitative examples of anomaly segmentations produced by AnomalyVFM. The examples can be seen in Figure 6. AnomalyVFM can detect anomalies across a wide range of objects.

G. Image Generation Data

To enable reproducibility and to ensure transparency, we provide the list of `[Object]`, `[Anomaly]` and `[Texture]` used in the synthetic dataset generation. The lists of `[Object]` and `[Anomaly]` tags can be seen in Table 4 and Table 5. The list of `[Texture]` tags can be seen in Table 6.

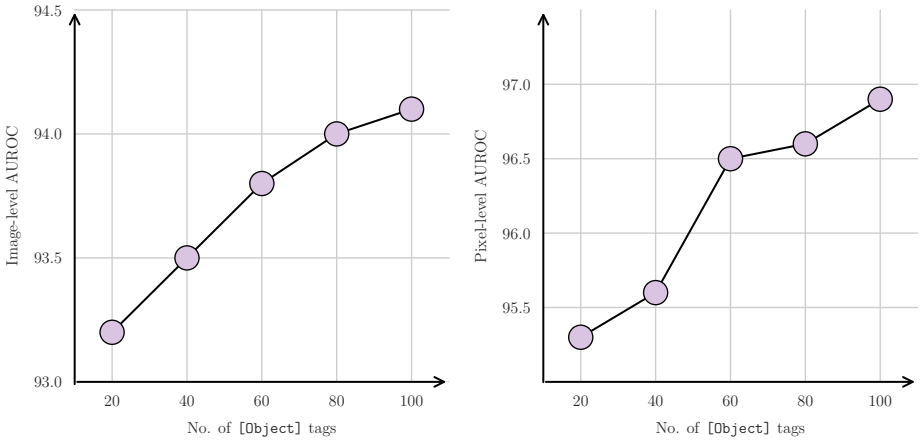

Figure 4. Model performance in comparison to the number of `[Object]` tags.

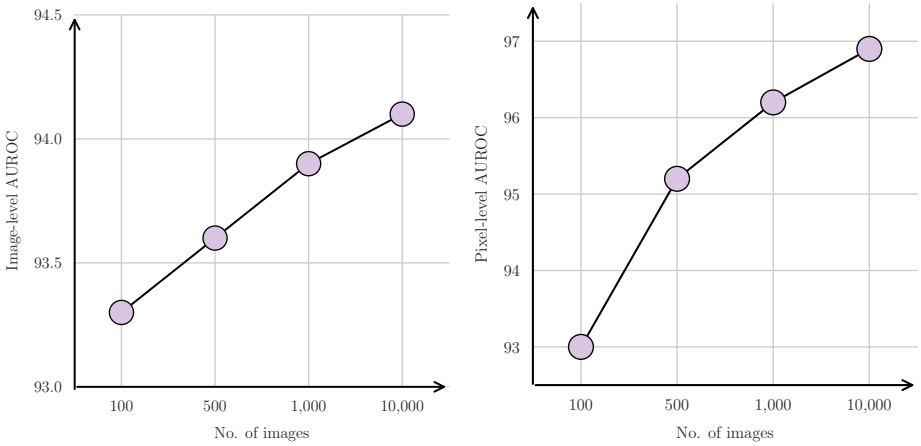

Figure 5. Model performance in comparison to the number of images in the training set.

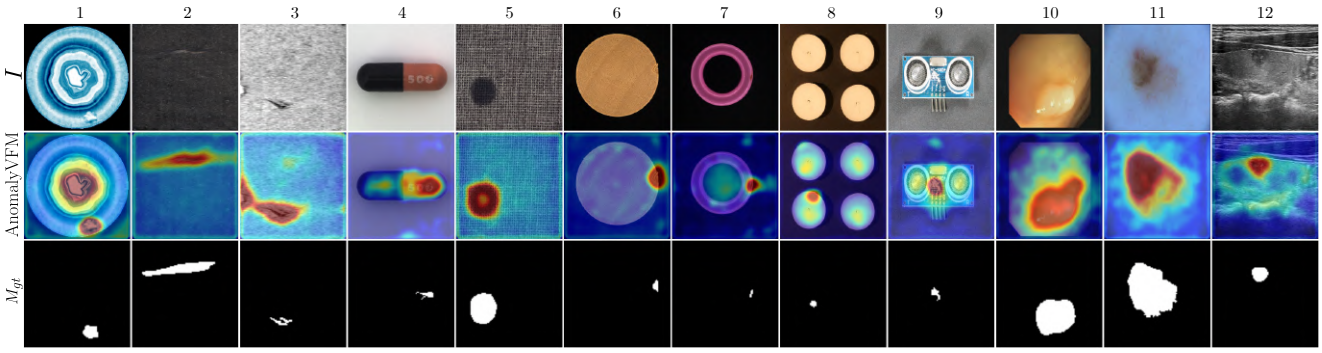

Figure 6. Qualitative examples of anomaly segmentation masks produced by AnomalyVFM. In the first row, the image is shown. In the next row, the anomaly segmentation produced by AnomalyVFM is depicted, and in the last row, the ground truth mask is depicted.

Table 4. [Object] and [Anomaly] data used for synthetic data generation. Here are listed objects from A to L.

[Object]	[Anomaly]
apple	[bruised, wrinkled, rotten, moldy, dented, discolored, soft spots]
apple slice	[oxidized, bruised, dried out]
asphalt	[cracked, pitted, faded, eroded, oil-stained, uneven]
ball	[deflated, scuffed, punctured, faded, cracked surface]
banana	[bruised, overripe, blackened, split peel, mushy, spotted]
battery	[leaking, corroded, dented, faded label]
belt	[cracked leather, frayed edges, worn holes, peeling finish]
bicycle	[flat tire, rusty chain, scratched frame, worn seat]
board game	[torn box, missing pieces, faded board, bent cards]
bread	[stale, moldy, crumbling, burnt, hardened, soggy]
brushed aluminum	[scratched, dented, stained, faded, oxidized, pitted]
butter	[rancid, melted, discolored, greasy residue, hardened]
car tire	[bald tread, cracked rubber, punctured, worn sidewall]
carbon fiber	[frayed, chipped, cracked, delaminated, scratched, discolored]
carrot	[softened, cracked, dehydrated, spotted, moldy, bent]
chair	[scratched wood, stained cushion, wobbly leg]
chalkboard	[scratched, smudged, cracked, chipped, stained, uneven]
cheese	[moldy, dried out, cracked, discolored, sweating, crumbly]
chocolate bar	[melted, bloomed, crumbled, discolored]
concrete	[cracked, pitted, stained, eroded, chipped, weathered]
cookies	[crumbled, stale, burnt, moldy]
cork	[cracked, crumbled, stained, dried out, warped, pitted]
corrugated metal	[dented, rusted, bent, scratched, corroded, pitted]
denim	[frayed, torn, stained, faded, pilled, worn]
doll	[torn clothing, missing eye, stained, frayed hair, loose limbs]
drill	[worn chuck, scratched casing, broken switch, dented battery]
egg	[cracked, leaking, discolored shell, dented, rotten, thin shell]
fabric	[frayed, torn, stained, faded, pilled, snagged]
fur	[matted, shedding, stained, torn, faded, dull]
garden hose	[cracked, leaking, kinked, faded]
garlic	[sprouted, dried out, moldy]
glasses	[scratched lenses, bent frame, loose arms, cloudy lenses]
gloves	[frayed fingers, stretched out, stained]
grape	[wrinkled, moldy, shriveled]
grill	[rusty grates, blackened residue, scratched body]
hammer	[rusty, chipped, bent, dented, scratched, loose head]
hat	[faded color, stretched, frayed edges]
headphones	[frayed cable, scratched ear cups, loose padding]
helmet	[scratched shell, cracked foam, loose straps]
hemp fabric	[frayed, torn, faded, pilled, stained, snagged]
jacket	[broken zipper, faded color, torn lining]
jeans	[worn knees, frayed hem, ripped pocket, faded]
key	[bent, worn teeth, rusty, scratched surface]
kite	[torn fabric, bent frame, frayed string, missing tail]
laminate	[scratched, chipped, peeled, bubbled, stained, warped]
lamp	[flickering, scratched base, broken switch]
laptop	[scratched casing, cracked hinge, faded keyboard keys]
lettuce	[wilting, yellowing, rotting]
light bulb	[burnt out, cracked, blackened, loose filament]
linen	[wrinkled, stained, faded, torn, frayed, pilled]

Table 5. [Object] and [Anomaly] data used for synthetic data generation. Here are listed objects from M to Z.

[Object]	[Anomaly]
mesh	[frayed, torn, snagged, discolored, brittle, stretched]
milk carton	[dented, leaking, stained, faded label, torn packaging]
mirror	[scratched, chipped edge, cloudy, stained surface]
onion	[sprouted, dried layers, rotting]
orange	[dried skin, moldy, bruised, discolored]
paintbrush	[frayed bristles, stiffened bristles, dried paint]
paper	[torn, wrinkled, stained, yellowed, brittle, moldy]
parquet flooring	[scratched, warped, faded, chipped, stained, dull]
phone	[cracked screen, scratched back, worn buttons]
plastic	[scratched, cracked, discolored, warped, brittle, faded]
pliers	[rusty, loose grip, scratched, chipped, stiff joint]
plywood	[warped, splintered, chipped, stained, delaminated, cracked]
potato	[sprouted, rotting, green spots, wrinkled, moldy, soft spots]
rake	[bent tines, rusty, loose handle]
rattan	[splintered, frayed, cracked, stained, brittle, discolored]
rubber floor	[cracked, brittle, discolored, stiffened, melted, torn]
saw	[rusty blade, dull teeth, chipped handle, bent blade]
scarf	[pilled fabric, snagged threads, stained]
screwdriver	[worn tip, rusty, scratched, bent, cracked handle]
shoes	[worn sole, scuffed leather, torn fabric, faded color]
shovel	[rusty blade, dented handle, worn grip]
smooth ceramic tile	[chipped, cracked, stained, crazed, dull, scratched]
smooth glass	[scratched, cracked, chipped, foggy, stained, shattered]
smooth metal	[rusted, scratched, dented, corroded, pitted, tarnished]
smooth wood plank	[cracked, splintered, warped, knotted, rotten, scratched, stained]
socks	[hole in toe, stretched elastic, faded]
stainless steel	[scratched, dented, stained, scuffed, fingerprinted, corroded]
stone tile	[chipped, cracked, eroded, stained, pitted, weathered]
strawberry	[moldy, bruised, shrinking]
synthetic fiber	[frayed, torn, stained, faded, pilled, melted]
table	[scratched surface, dented corner, water stains]
tape measure	[cracked casing, faded markings, stuck mechanism]
teddy bear	[ripped seam, matted fur, faded color, stained, missing stuffing]
tent	[torn fabric, bent poles, moldy spots]
tomato	[soft spots, cracked skin, moldy]
toy car	[scratched paint, missing wheel, cracked body, loose parts]
TV remote	[worn-out buttons, cracked case, faded labels]
velvet	[crushed, faded, stained, pilled, torn, frayed]
wallet	[worn edges, cracked leather, faded color, frayed stitching]
wallpaper	[peeled, torn, stained, faded, bubbled, wrinkled]
watch	[scratched face, broken strap, faded markings, cracked casing]
whiteboard	[scratched, stained, ghosting, cracked, faded, dented]
window	[scratched glass, cracked, foggy]
woven mat	[frayed, torn, faded, loose fibers, stained, worn]
wrench	[rusty, scratched, dented, worn edges, corroded]
yo-yo	[scratched, cracked, tangled string, chipped edge]

Table 6. [Texture] data used for synthetic dataset generation.

[Texture]
asphalt, **bamboo**, **brick**, **brushed aluminum**, **canvas**
carbon fiber, **ceramic**, **chalkboard**, **clouds**, **concrete**
cork, **corrugated metal**, **denim**, **fabric**, **fleece**
foam, **fur**, **glass**, **granite**, **grass**
gravel, **hemp fabric**, **ice**, **laminate**, **linen**
marble, **mesh**, **metal**, **mirror**, **painted wall**
paper, **parquet flooring**, **pebbles**, **plastic**, **plywood**
rattan, **rubber**, **sand**, **snow**, **stainless steel**
stone, **synthetic fiber**, **tarpaulin**, **terrazzo**, **tile**
velvet, **wallpaper**, **whiteboard**, **wire mesh**, **woven mat**

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD–A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 2

[2] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. AdaCLIP: Adapting CLIP with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024. 1

[3] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1

[4] Wenxin Ma, Xu Zhang, Qingsong Yao, Fenghe Tang, Chenxu Wu, Yingtai Li, Rui Yan, Zihang Jiang, and S Kevin Zhou. Aa-clip: Enhancing zero-shot anomaly detection via anomaly-aware clip. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4744–4754, 2025. 1

[5] Zhen Qu, Xian Tao, Xinyi Gong, ShiChen Qu, Qiyu Chen, Zhengtao Zhang, Xingang Wang, and Guiguang Ding. Bayesian Prompt Flow Learning for Zero-Shot Anomaly Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 30398–30408, 2025. 1

[6] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1

[7] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1

[8] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection. In *ICLR*, 2024. 1

[9] Jiawen Zhu, Yew-Soon Ong, Chunhua Shen, and Guansong Pang. Fine-grained abnormality prompt learning for zero-shot anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22241–22251, 2025. 1