SQUINT YOUR EYES: CHOOSING WHEN TO BLUR FOR SPARSE VISUAL DOMAINS

Anonymous authors
Paper under double-blind review

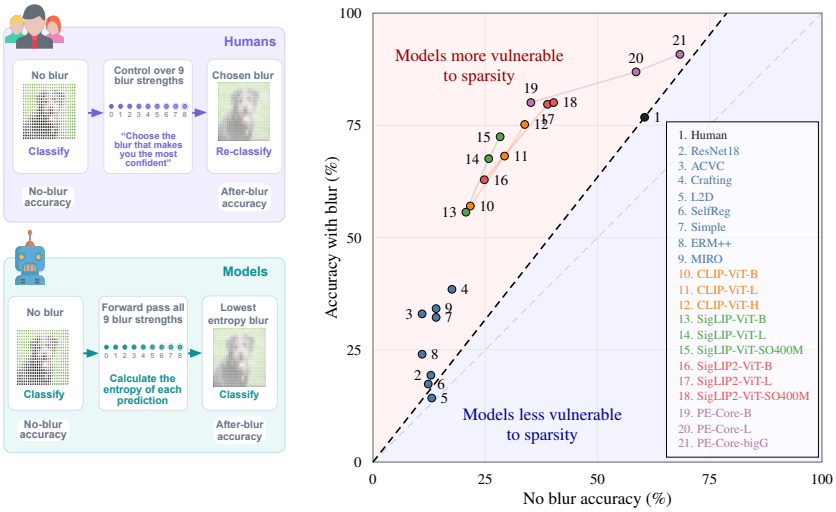

Figure 1: **Vulnerability to sparsity for humans and machines. Left:** Evaluation protocol for our human study and model comparison. We study images in which object information is represented by spatially disconnected elements. Humans first classify the image without blur, then choose among nine blur levels the version that maximizes their confidence, and classify again. Models follow the same protocol, using the blur level with the lowest prediction entropy. **Right:** Results of the human vs model comparison. Accuracy without blur (x-axis) versus accuracy with selected blur (y-axis). The black dashed line passes through the human point and represents the human blur/no-blur accuracy ratio. Nearly all models are above this line, benefiting more from blur than humans do.

ABSTRACT

Visual recognition is typically studied under shifts in style or corruption, but the effect of spatially disconnected visual evidence remains underexplored. We study visual sparsity, where objects are depicted as separate dots, symbols, or line fragments. Motivated by the human tendency to squint when encountering fragmented visual patterns, we investigate whether blurring helps recognition under sparsity. We conduct a controlled human study and an analogous model evaluation. Blur improves recognition under sparsity for both humans and modern recognition models, but the relative improvement is substantially larger for models, indicating that they are more vulnerable to sparsity. This human–model mismatch creates an exploitable gap for moderation attacks, where sparse images can remain recognizable to people yet be substantially harder for moderation systems. We demonstrate this failure mode in a content-moderation setting. To study sparsity systematically, we introduce **DN AbstraGrid**, a benchmark based on Mini DomainNet, that spans multiple abstraction levels: a sparse target partition, a non-sparse partition, and the original domains. The latter two serve as control domains where performance should be preserved. Blur strongly benefits the sparse partition while harming the remaining domains, showing that it should be applied adaptively rather than uniformly. To this end, we propose **Squint!**, a lightweight, model-agnostic module that predicts when and how much to blur each image. It is based on a compact radial Fourier power-spectrum descriptor, trained with weak supervision and not on the target sparse domains. Squint! improves recognition on sparse domains while preserving performance on the controls. A single plug-and-play Squint! model improves performance across 12 VLMs and 7 domain-generalization methods, while also improving recognition of sparse moderation attacks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Domain shift in visual recognition is commonly studied under changes in style Li et al. (2017), texture Geirhos et al. (2019), or corruption Hendrycks & Dietterich (2019). Less attention has been paid to domains in which the visual evidence composing an object becomes spatially disconnected. The human visual system can organize discrete visual elements into coherent shape representations Baker & Kellman (2024), and this ability extends to semantic recognition Panis et al. (2008). This suggests that sparse visual evidence can remain perceptually meaningful despite fragmentation. In this paper, we examine recognition in the setting of *visual sparsity*, where an object is represented by spatially disconnected components such as dots, symbols, or line fragments separated by gaps. Importantly, sparsity here does not mean low pixel occupancy or thinness. A continuous outline, for example, is not sparse under this definition. Sparsity introduces a high-frequency signal unrelated to the content. Humans, beyond their passive perceptual ability, adopt simple behavioral strategies to suppress high frequencies in order to improve recognition under sparsity. This includes squinting or viewing the image from a distance. As blur is a simple high-frequency removal method, we examine its effect on recognition accuracy of humans and current deep models in visually sparse domains.

We conduct a controlled human study, where participants first classify sparse images, and then they repeat the task after being given access to a predefined set of blur levels. This set-up allows us to test whether suppressing fine-scale detail improves recognition. We perform an analogous evaluation across modern recognition models, selecting the blur level with the lowest prediction entropy. Blur significantly improves human recognition, but the relative improvement is substantially larger for most evaluated models. We quantify this effect using the blur/no-blur accuracy ratio. We focus on this relative effect because absolute accuracy is highly sensitive to the task definition, including the number of classes, the level of class granularity, and the amount of domain-specific knowledge required to distinguish them. The blur/no-blur ratio instead measures how strongly sparsity affects each recognizer relative to its own baseline. This reveals that models are generally more vulnerable than humans to spatially disconnected visual evidence (Figure 1), creating a practical blind spot. We further demonstrate this gap in content moderation, where sparse transformations of hate symbols substantially reduce automated recognition while remaining recognizable to humans.

To study this regime systematically, we introduce **DN AbstraGrid**, a benchmark built on Mini DomainNet that re-renders each image through eight abstraction families at ordered levels of abstraction. While HAID Li & Jiao (2025) proposes ordered abstraction within a single non-sparse family, DN AbstraGrid contains four *sparse* re-rendering families, which form our target partition, and four *non-sparse* re-rendering families, which serve as control domains. We also retain the original Mini DomainNet as a second control set. This design lets us evaluate whether a method can improve sparse recognition while preserving performance on standard images and non-sparse abstractions.

Blur improves recognition on sparse domains, but applying it uniformly is not viable because it harms recognition on non-sparse inputs (Figure 2). The question is therefore *when to blur and by how much*. We address this with **Squint!**, a lightweight, model-agnostic pre-processing module that selects a blur strength from a compact radial Fourier power-spectrum descriptor and applies it before passing the image to any downstream model. Squint! is trained with weak supervision and without using the target sparse domains, showing that its blur selection generalizes to unseen sparsity.

A single plug-and-play Squint! improves recognition across 12 Vision-Language Models (VLMs) and 7 Domain Generalization (DG) methods while preserving performance on the control domains, and can reduce multi-blur inference cost by pruning unnecessary blur levels. The same model also improves recognition of sparse moderation attacks without disrupting ordinary moderation.

Our contributions are:

- We identify **visual sparsity** as an underexplored recognition regime and introduce **DN AbstraGrid**, a benchmark designed to study sparsity systematically.
- Through a **controlled human study**, we show that blur improves the recognition of sparse images for both humans and models, but that models benefit disproportionately more.
- We show that this human–model mismatch creates a **practical moderation blind spot**, in which sparse obfuscations can remain human-recognizable yet be harder for models.
- We propose **Squint!**, a lightweight, model-agnostic blur selector demonstrated on 12 VLMs, 7 domain generalization methods, and on a realistic moderation attack scenario.

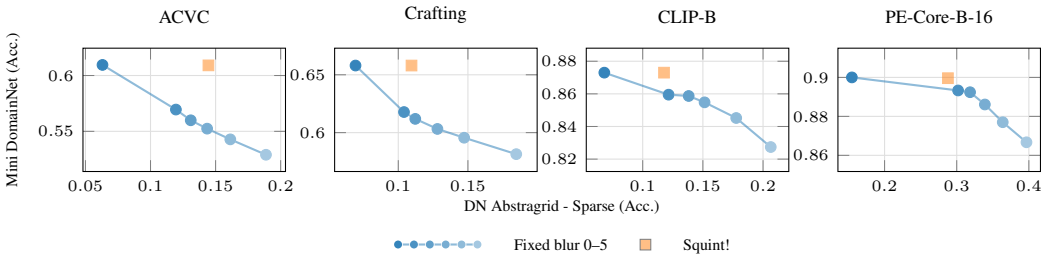

Figure 2: **DN AbstraGrid sparse vs. Mini DomainNet performance trade-off:** Fixed Gaussian blur levels ($\sigma = 0, 1.1, 1.3, 1.5, 1.7, 2.1$) improve accuracy on sparse domains but rapidly degrade performance on Mini DomainNet. Squint! adaptively decides which images should be blurred.

2 Related Work

Domain Generalization. A major line of DG work reduces texture reliance and promotes shape bias Geirhos et al. (2019). This includes random-convolution methods Xu et al. (2021); Choi et al. (2023) and edge-based methods that use either a trainable HED Xie & Tu (2017) detector in Harary et al. (2022) or a Canny Canny (1986)-based edge augmentation Efthymiadis et al. (2022) in Efthymiadis et al. (2025). Other methods use self-supervised contrastive regularization Kim et al. (2021) or corruption-based attention consistency Cugu et al. (2022). These approaches target domain shifts where focusing on shape improves robustness. The sparse domains studied here instead break object structure into disconnected elements, making the underlying shape harder to recover. We evaluate 7 DG methods in this regime, all of which are improved by Squint!.

Human Recognition. Human object recognition is remarkably robust to changes in visual representation. Simplified line drawings can support recognition nearly as effectively as photographs Biederman & Ju (1988), and humans remain robust across photographs, drawings, and sketches even as CNN performance degrades with increasing abstraction Singer et al. (2022). Humans also rely more on shape cues than standard ImageNet-trained CNNs Geirhos et al. (2019) and use a narrower range of spatial frequencies for object recognition Subramanian et al. (2023). Spatially separated elements can be interpreted as coherent shape representations Baker & Kellman (2024), and this ability extends to semantic recognition of fragmented outlines Panis et al. (2008); Scialom et al. (2026). Low-spatial-frequency information also contributes to object recognition and early hypotheses about object identity Bar et al. (2006); Fintzi & Mahon (2014). Blur has also been explored as a way to bring model behavior closer to human recognition. Training with clear and blurred images improves human–model alignment Jang & Tong (2024); Yoshihara et al. (2023), while test-time low-pass filtering can improve error consistency and shape bias at the cost of recognition accuracy Wolff et al. (2026). We instead study how humans and models differ under visual sparsity, whether blur reduces this gap, and how to apply it selectively without harming recognition when unnecessary.

Datasets and Benchmarks. Most DG benchmarks use discrete domains such as photo, sketch, or painting, as in PACS Li et al. (2017), DomainNet Peng et al. (2019), and Office-Home Venkateswara et al. (2017). They capture abstraction only coarsely, since a sketch can be more abstract than a photograph, but each domain has a fixed level of abstraction. Corruption benchmarks such as CIFAR-10C and ImageNet-C Hendrycks & Dietterich (2019) provide ordered severity, but model degradation rather than abstraction as a careful re-rendering that respects recognition. HAID Li & Jiao (2025) introduces ordered abstraction levels, but only within a single non-sparse family. DN AbstraGrid extends this to eight families, with four sparse target domains and four non-sparse controls. Finally, Image Obfuscation Benchmark Stimberg et al. (2023) is an adversarial image obfuscation dataset that challenges recognition systems. While it is not designed to study visual sparsity, its distortions span a wide frequency range, making it a suitable choice for training Squint!.

Fourier Power Spectrum. Fourier-domain statistics have long been used to characterize image structure and scene properties Field (1987); Oliva et al. (1999); Torralba & Oliva (2003). Radially averaged power spectra provide a compact, orientation-agnostic summary of frequency content Field (1987) and have also been used for no-reference blur assessment Mavridaki & Mezaris (2014). Squint! builds on this idea by representing each image with a log-compressed radial histogram of its Fourier power spectrum. Rather than using these statistics to characterize images or estimate blur, we use them to predict when blur can be beneficial.

3 THE DN ABSTRAGRID BENCHMARK

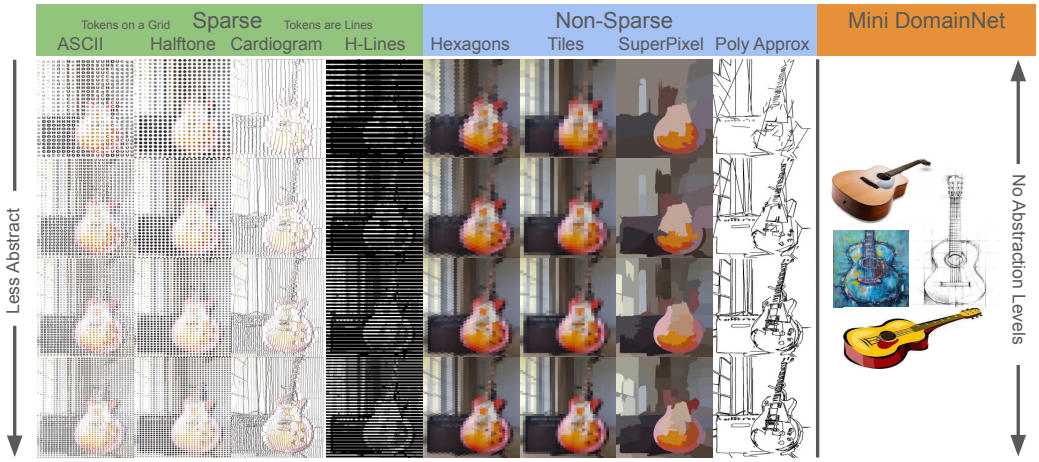

Figure 3: **DN AbstraGrid.** Overview of the eight abstraction families with ordered abstraction.

Controlled Visual Abstraction. DN AbstraGrid is built on Mini DomainNet and re-renders the same images through eight abstraction families, each at four ordered levels. Following HAID Li & Jiao (2025), we model abstraction strength explicitly rather than representing each abstract domain at a single level. While HAID proposes a single non-sparse abstraction family, DN AbstraGrid includes eight, four of which are sparse. We distinguish abstraction from corruption. Corruption degrades an existing depiction, whereas abstraction changes how visual content is represented while aiming to preserve semantics. Our domains are drawn from established conventions in art, print, and digital media, where such representations have long been used to depict recognizable content.

The Domains of DN AbstraGrid. DN AbstraGrid distinguishes domains by whether visual support is spatially disconnected. We define a depiction as sparse when its visual evidence is distributed across disconnected components separated by gaps. Sparsity, therefore, does not mean low pixel occupancy. A thin but connected outline is not sparse under this definition. Four families, *ASCII*, *Halftone*, *Cardiogram*, and *H-Lines*, form the sparse target set. The remaining four, *Hexagons*, *Tiles*, *SuperPixel*, and *Polygonal Approximation*, form the non-sparse control set. Together with the original Mini DomainNet, which shares the same semantic classes, they serve as controls on which performance should be preserved. An overview of DN AbstraGrid is shown in Figure 3.

In and Out of Domain Control Sets. The original Mini DomainNet provides an *in-domain control*. Its validation split is available during method development as a preservation filter, rejecting methods that harm standard-domain performance. The final evaluation uses the disjoint test split. The non-sparse DN AbstraGrid partition provides an *out-of-domain control* and tests whether gains on sparse domains preserve performance on unseen abstractions. A successful method should improve the sparse targets while preserving both controls. Full rendering details are provided in Appendix A.1.

4 HUMAN STUDY

Protocol and Metric. We test whether blur aids human recognition of sparse depictions and whether humans and models are affected by sparsity to the same extent. We conduct a controlled study with 16 adult participants using 16-class ImageNet images re-rendered with the four sparse re-renderings of DN AbstraGrid. Participants first classify each image, then view the same image under nine blur levels, select the version for which they are most confident, and classify it again. Models follow the analogous protocol using the blur level with the lowest entropy. As a metric, we use the blur/no blur accuracy ratio. Unlike absolute accuracy, which depends on task difficulty, this ratio measures each recognizer relative to its own baseline. Most evaluated models have substantially higher ratios, indicating greater sensitivity to sparsity. The procedure is shown in Figure 1.

Effect of Blur. Blur substantially improves human recognition, increasing accuracy from 60.5% to 76.8%. All 16 participants improve after blur, and the effect is significant under an exact one-sided participant-level sign test ($p = 1.53 \times 10^{-5}$). More details in Appendix A.2.

5 METHOD

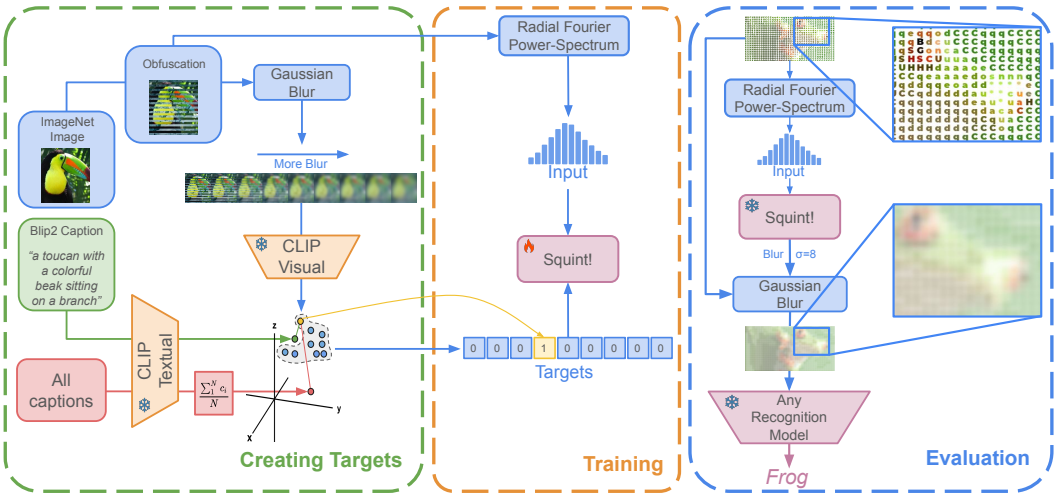

Figure 4: **Training and evaluation of Squint!.** During training, weak supervision is derived from image–caption alignment. At inference time, the module operates only on a compact Fourier descriptor of the image to propose a blur level if needed. Any recognition model can be used.

To improve recognition on sparse domains without harming performance on non-sparse domains, we train a blur-level selection model using the procedure illustrated in Figure 4 (left and middle). Squint! is a lightweight, model-agnostic pre-processing module that adaptively selects and applies a discrete blur level to each input before it is passed to a downstream recognition model. Squint! is trained without access to the sparse domains of **DN AbstraGrid**. Since most images encountered in realistic settings are expected to be non-sparse, a key requirement is to preserve recognition performance on the non-sparse control domains. We enforce this through a validation-based model selection step that filters out Squint! models that degrade control-domain accuracy. Although training and model selection are developed using an off-the-shelf VLM, the resulting Squint! can serve as a plug-and-play pre-processor for any recognition model without further fine-tuning.

Training Data. Squint! is trained on 250,000 image–caption pairs derived from the Image Obfuscation Benchmark Stimberg et al. (2023). For each selected obfuscated image, we retrieve its corresponding clean ImageNet Deng et al. (2009) image and generate a caption for the clean image using BLIP-2 Li et al. (2023a). The resulting training pairs, therefore, consist of an obfuscated image paired with a caption describing its clean counterpart. We deliberately use captions as supervision to demonstrate that Squint! can be trained with weak supervision, without class-level labels.

Radial Fourier Power-Spectrum Descriptor. For each augmented image, a fixed-length descriptor is computed to summarize how image energy is distributed across spatial frequencies. The image is converted to grayscale, intensities are normalized to $[0, 1]$, the 2D Fourier transform is computed, and the power spectrum $P = |F|^2$ is formed. To obtain a rotation-invariant summary, the spectrum is averaged over concentric rings in the frequency plane using uniform radial bins from zero frequency to the maximum discrete 2D frequency radius, followed by a log transform for dynamic-range compression. This produces a 64-D vector used as input to Squint!.

Blur Candidate Set. Blur is discretized into $B = 9$ ordered candidate levels. The choice of Gaussian blur over other frequency-suppression strategies and the number of blur candidates are motivated by Figure 7 and Figure 8 of the Appendix. We parameterize blur strength through effective resolution. For the native model resolution $N = 224$, an effective resolution r_i corresponds to the relative spatial scale $S_i = N/r_i$. In Gaussian pyramids, filter width grows proportionally as resolution decreases, while Gaussian scale-space similarly gives $\sigma \propto S$ under spatial scaling Burt & Adelson (1983); Lindeberg (1996). We use a unit-pixel reference scale and set $\sigma_i = 224/r_i$ for effective resolutions r_i linearly spaced from 224 to 34. The first candidate is the identity transform.

Weak Targets from Caption-Specific CLIP Alignment. To obtain supervision without class labels, we use CLIP Radford et al. (2021) with a ViT-B/16 Dosovitskiy et al. (2021) backbone to select, for each image–caption pair, the blur level that best aligns the obfuscated image with the cap-

tion of its clean counterpart. For each candidate blur, we compute the cosine similarity between the image and caption embeddings. Simply maximizing this similarity can favor blur levels that move the image embedding toward the text manifold in general rather than toward its specific caption. We therefore normalize the image–caption similarity by the image embedding's distance from the mean text embedding over all 250,000 training captions. This caption-specific alignment correction favors blur levels that improve caption-specific alignment rather than generic proximity to the text modality. Each pair is assigned a one-hot target corresponding to the highest-scoring blur level.

Squint! Architecture. Squint! is implemented as a small MLP that maps the 64-D Fourier descriptor to a distribution over $B = 9$ blur levels. The descriptor is first passed through LayerNorm Ba et al. (2016), followed by a fully connected hidden block with LayerNorm, ReLU, dropout Srivastava et al. (2014), and a residual connection He et al. (2016). A final linear layer maps to B outputs, followed by LayerNorm, dropout, and a softmax over blur candidates.

Training Objective. We train Squint! with a cross-entropy loss L_{CE} on the radial Fourier power-spectrum descriptors and the weakly supervised blur labels. To avoid collapsing to trivially predicting the no-blur class, a diversity regularizer L_E is added that maximizes the entropy of the batch-average prediction distribution. This encourages multiple blur outputs. The total objective is:

$$L_{\text{total}} = L_{\mathrm{CE}} - \lambda L_E$$

Model Selection. We select a single Squint! model prior to evaluation on DN AbstraGrid. Candidate models are ranked by their blur-target prediction performance and then filtered using the Mini DomainNet *validation split*, retaining only models whose selected blur does not reduce recognition accuracy on the original real, clipart, painting, and sketch domains. Among these, we choose the highest-ranked candidate, falling back to the overall best-ranked model if necessary. DN AbstraGrid is constructed exclusively from the Mini DomainNet *test split*, including both the original images and their re-rendered abstractions, ensuring that model selection is fully disjoint from evaluation.

Using Squint! at Inference. We use Squint! in two complementary modes. *Single-pass* applies the blur level proposed by Squint! and evaluates only the resulting image, requiring a single forward pass through the downstream recognizer. If multi-blur ensemble recognition is preferred, Squint! can instead be used to prune the candidate set. Its proposed blur acts as a lower bound, excluding weaker blur levels so that only the remaining candidates are evaluated. This reduces the number of forward passes while further improving ensemble recognition on sparse domains.

6 Experiments

Evaluation Setup. Table 1 summarizes our main results across DG and VLMs. The three-column groups correspond to DN AbstraGrid Sparse, DN AbstraGrid Non-Sparse, and the original Mini DomainNet domains. Our goal is to improve the sparse partition without harming either of the control partitions. *No blur* denotes the original model prediction, while *Squint!* applies the proposed blur before a single forward pass through the corresponding model. *MC-E* performs multi-blur recognition by evaluating all nine candidates and selecting the one with the lowest entropy. *Squint! + MC-E* uses the Squint! proposal as a lower bound, pruning weaker blur levels before ensembling. This reduces the number of forward passes while further improving recognition on sparse domains.

Domain Generalization. We evaluate Squint! on a ResNet18 He et al. (2016) baseline and seven DG methods. L2D Wang et al. (2021b), ACVC Cugu et al. (2022), Crafting Efthymiadis et al. (2025), and SelfReg Kim et al. (2021) follow the single-source setting, trained on Mini DomainNet Photo with a ResNet18. MIRO Cha et al. (2022), SIMPLE Li et al. (2023b), and ERM++ Teterwak et al. (2025) use multi-source training on all four Mini DomainNet domains with a ResNet50. DN AbstraGrid Sparse provides a reality check: single-source methods reach only 2.6–6.9% accuracy, and multi-source methods only 2.1–2.9%, exposing robustness gaps beyond the non-sparse shifts typically studied in DG. Squint! consistently improves sparse recognition, nearly doubling average accuracy from 3.7% to 7.3%. Combined with MC-E, Squint! increases accuracy from 11.9% to 12.8% while requiring fewer forward passes. Performance on both controls is preserved.

Vision-Language Models. We further evaluate Squint! across 12 VLMs from the CLIP Radford et al. (2021), SigLIP Zhai et al. (2023), SigLIP2 Tschannen et al. (2025), and PE-Core Bolya et al. (2025) families, spanning multiple model scales and input resolutions. Squint! is applied in a

Method	DN AbstraGrid sparse (To improve)				DN AbstraGrid non-sparse (Control dataset 1)				Mini DomainNet (Control dataset 2)			
	No blur	Squint! (ours)	MC-E	Squint! + MC-E (ours)	No blur	Squint! (ours)	MC-E	Squint! + MC-E (ours)	No blur	Squint! (ours)	MC-E	Squint! + MC-E (ours)
ResNet-18	2.6	4.1↑	5.5	6.3↑	13.1	13.1✓	22.8	22.8✓	55.0	55.0✓	52.4	52.4✓
L2D	3.0	5.0↑	6.1	6.7↑	22.4	22.4✓	28.0	28.0✓	59.5	59.5✓	57.5	57.5✓
ACVC	6.3	14.4↑	27.4	29.0↑	44.8	44.8✓	49.6	49.6✓	61.0	60.9✗	57.5	57.4✗
Crafting	6.9	10.9↑	20.1	21.8↑	51.1	51.1✓	51.9	51.9✓	65.8	65.8✓	63.6	63.6✓
SelfReg	3.1	4.7↑	5.3	5.6↑	16.3	16.3✓	18.9	18.9✓	59.6	59.6✓	54.2	54.2✓
MIRO	2.9	7.5↑	11.5	12.2↑	18.2	18.1✗	30.5	30.4✗	99.7	99.7✓	99.7	99.7✓
SIMPLE	2.5	7.5↑	11.7	12.8↑	19.3	19.3✓	28.5	28.5✓	99.3	99.3✓	98.5	98.5✓
ERM++	2.1	4.7↑	7.6	8.3↑	16.3	16.3✓	24.2	24.2✓	99.6	99.6✓	99.0	99.0✓
Average DG	3.7	7.3↑	11.9	12.8↑	25.2	25.2✓	31.8	31.8✓	74.9	74.9✓	72.8	72.8✓
CLIP-ViT-B-16	6.8	11.8↑	29.4	29.9↑	37.8	37.8✓	52.8	52.8✓	87.3	87.3✓	86.6	86.6✓
+ fine-tuned blur	6.9	12.0↑	31.4	31.7↑	38.3	38.3✓	54.0	54.0✓	87.3	87.3✓	86.4	86.4✓
+ fine-tuned imgaug	8.8	14.3↑	35.3	35.7↑	46.9	47.0✓	57.1	57.1✓	87.6	87.6✓	86.9	86.9✓
+ fine-tuned full	8.9	14.4↑	36.1	36.5↑	47.5	47.6✓	57.2	57.2✓	87.5	87.5✓	86.8	86.8✓
CLIP-ViT-L-14	11.2	16.3↑	35.5	35.9↑	45.6	45.5✗	57.9	57.9✓	90.1	90.1✓	88.8	88.8✓
CLIP-ViT-H-14	11.4	19.2↑	37.0	37.8↑	50.6	50.7✓	60.2	60.2✓	90.8	90.7✗	89.8	89.8✓
SigLIP-ViT-B-16	12.0	19.8↑	33.1	33.3↑	49.8	49.8✓	56.5	56.5✓	91.1	91.1✓	90.6	90.6✓
SigLIP-ViT-L-16-384	17.1	18.6↑	42.3	42.4↑	57.7	57.7✓	64.4	64.4✓	93.2	93.2✓	93.0	93.0✓
SigLIP-ViT-SO400M-14-384	18.0	22.1↑	41.9	42.3↑	59.4	59.3✗	65.3	65.3✓	93.7	93.7✓	93.5	93.5✓
SigLIP2-ViT-B-16-256	13.6	19.7↑	37.3	37.8↑	53.2	53.2✓	60.4	60.4✓	92.1	92.1✓	91.5	91.5✓
SigLIP2-ViT-L-16-256	21.8	28.1↑	50.1	50.5↑	59.3	59.3✓	64.7	64.7✓	93.2	93.2✓	92.8	92.8✓
SigLIP2-ViT-SO400M-16-384	21.3	25.9↑	51.3	51.4↑	61.4	61.4✓	66.7	66.7✓	93.7	93.7✓	93.4	93.4✓
PE-Core-B-16	15.5	28.8↑	47.7	48.1↑	57.7	57.7✓	62.1	62.1✓	90.0	90.0✓	89.0	89.0✓
PE-Core-L-14-336	32.7	37.3↑	59.4	59.7↑	68.7	68.7✓	70.3	70.3✓	92.3	92.3✓	92.5	92.5✓
PE-Core-bigG-14-448	46.4	47.6↑	69.1	69.2↑	74.5	74.5✓	75.9	75.9✓	93.8	93.8✓	93.7	93.7✓
Average VLM	16.8	22.4↑	42.5	42.8↑	53.9	53.9✓	61.7	61.7✓	90.9	90.9✓	90.3	90.3✓

Table 1: **Main results.** Squint! improves recognition on DN AbstraGrid Sparse while preserving performance on the non-sparse and Mini DomainNet controls. A green arrow (↑) denotes an improvement on sparse domains, a green check (✓) denotes preserved control performance, and a red cross (✗) denotes a decrease on a control. Across individual backbones, any control degradation is at most 0.1%, while most configurations preserve control accuracy and the averages remain unchanged. Results include DG and VLM models for both single-pass Squint! and Squint! + MC-E.

plug-and-play manner across all 12 VLMs, without retraining or adaptation. It improves average accuracy on DN AbstraGrid Sparse from 16.8% to 22.4% while preserving accuracy on both control partitions. Squint! remains complementary to MC-E, increasing average sparse accuracy from 42.5% to 42.8% while pruning blur levels and reducing the number of forward passes.

Single-pass vs. multi-blur inference. MC-E provides larger gains on sparse domains than single-pass Squint!, but at a substantially different accuracy–computation trade-off. Across DG models, MC-E raises average sparse accuracy from 3.7% to 11.9%, while reducing Mini DomainNet accuracy from 74.9% to 72.8%. Single-pass Squint! reaches 7.3% while preserving the original 74.9% accuracy. The same trend holds for VLMs, where MC-E reduces Mini DomainNet accuracy from 90.9% to 90.3%, while Squint! preserves 90.9%. Preserving accuracy on non-sparse inputs is particularly important, as sparse images are expected to occur less frequently in real-world scenarios.

MC-E also evaluates all blur levels, making its cost scale drastically with the recognition backbone. For CLIP-B, computation increases from 35.1 to 316.4 GFLOPs/image, while for CLIP-H it increases from 334.7 to 3012.3 GFLOPs/image. Single-pass Squint! instead adds only its lightweight blur prediction before one recognition forward pass. MC-E provides higher sparse accuracy at the cost of reduced performance on the original domains and multiple forward passes. When multi-blur inference is preferred, Squint! can prune the MC-E candidate set, further improving sparse accuracy while reducing the number of forward passes.

Complementarity with robust fine tuning. We next test whether improving the robustness of the recognition model itself makes adaptive blur unnecessary. Starting from CLIP ViT B/16 pretrained on LAION 2B Schuhmann et al. (2022), we fine-tune it on LAION 2B using either the ImgAug [1] augmentation family, multiple levels of Gaussian blur, or both. Each component improves recognition of sparse images. No blur accuracy increases from 6.8% to 6.9% with blur training, 8.8% with ImgAug, and 8.9% with both. The corresponding Squint! results also improve from 11.8% to 12.0%, 14.3%, and 14.4%, respectively. Making the recognition model more robust does not render Squint! redundant. Instead, backbone robustness and adaptive blur provide complementary gains.

Supervised variant. We also test whether the benefits of Squint! rely on the weakly supervised target-construction pipeline based on CLIP and BLIP-2. We train a supervised variant using ImageNet class labels and an ImageNet-trained ViT-B classifier, while keeping the Fourier representation, Squint! architecture, and blur candidates unchanged. Only the construction of the blur targets

[1] https://github.com/aleju/imgaug

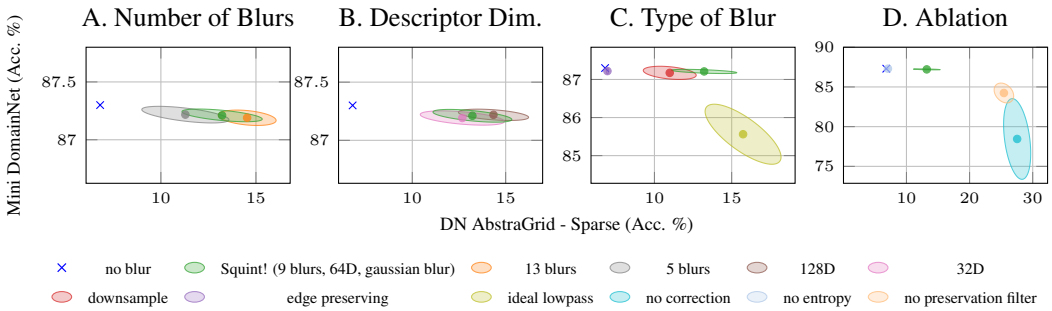

Figure 5: **Impact of parameters and ablation study of Squint!** Each scatter plot compares average accuracy on DN AbstraGrid-Sparse (x-axis) against Mini DomainNet (y-axis). Ellipses show one-standard-deviation contours of fitted multivariate Gaussians, estimated by repeating each experiment 20 times. Center marks mean performance. The default configuration, Squint! with 9 Gaussian blur levels and 64-bin histograms, is shown alongside no blur and each ablated setting.

is modified. Rather than selecting the blur level that maximizes CLIP alignment with the caption, we evaluate the ViT-B classifier across all blur candidates and select the lowest blur level at which it predicts the correct ImageNet class. Samples for which no blur level recovers the correct class are excluded from training. For direct comparison with the weakly supervised variant, we evaluate the resulting Squint! model using the same CLIP ViT-B/16 recognition backbone at test time. Despite changing both the model used to construct the targets and the supervision signal, the supervised variant improves sparse accuracy from 6.8% to 9.4% and MC-E from 29.4% to 29.7%, while preserving accuracy on both controls.

Impact of Parameters and Ablations. We study the main design choices of Squint! in Figure 5, using the trade-off between accuracy on DN AbstraGrid Sparse and the Mini DomainNet control set. Each ablation is repeated 20 times to account for training variability. The contours correspond to the one-standard-deviation level of the fitted two-dimensional Gaussian distributions. The number of blur candidates (*A. Number of Blurs*) and the dimensionality of the Fourier descriptor (*B. Descriptor Dim.*) are both stable choices, with the tested configurations lying within approximately one standard deviation of one another. We use 9 blur levels and a 64-dimensional descriptor as a representative intermediate configuration rather than tuning these choices aggressively. The choice of frequency-suppression strategy (*C. Type of Blur*) is important. For each strategy, candidate levels are formed by varying a single strength parameter. Downsampling lowers the intermediate resolution before resizing back, edge-preserving filtering increases the bilateral-filter neighborhood, and ideal low-pass filtering lowers the radial Fourier cutoff. Downsampling yields smaller gains on sparse domains, while edge-preserving filtering is largely ineffective. Ideal low-pass filtering can substantially improve sparse recognition, but none of its 20 runs passes the Mini DomainNet preservation filter, so we report the fallback models described in Model Selection. Gaussian blur provides the best trade-off, consistent with the fixed-blur study in Figure 7 of the Appendix. We next isolate the components of Squint! (*D. Ablation*). Removing the entropy regularizer eliminates the gains on sparse domains, while removing the caption-specific alignment correction causes all 20 runs to fail the Mini DomainNet preservation criterion. Finally, we deliberately disable the preservation filter to examine its role directly. This yields higher sparse accuracy, but at the cost of a large drop on Mini DomainNet, showing the importance of explicitly preserving performance on the original domains.

Latency. Squint! requires fewer than 6 MFLOPs per image. For comparison, a ResNet18 forward pass requires 7.3 GFLOPs, while CLIP-B requires 35.1 GFLOPs. The computational cost of Squint! is therefore several orders of magnitude smaller than that of the downstream model, so it is negligible in practice. For sparse inputs, Squint! also reduces the cost of MC-E by pruning unnecessary blur levels before recognition, requiring fewer backbone forward passes while retaining the benefit of multi-blur inference. Detailed latency measurements are provided in Figure 10 of the Appendix.

Test-Time Adaptation. We next ask whether standard test-time adaptation can address recognition under sparse domain shifts. We compare against MEMO Zhang et al. (2022) and TENT Wang et al. (2021a), applying each independently to every domain generalization backbone. MEMO operates on individual test samples and therefore matches our setting without assumptions about the batch. TENT instead adapts over batches, so we evaluate it using batches of 128 images containing 1%–50% DN AbstraGrid Sparse samples, with the rest drawn from Mini DomainNet. We treat 1% as

Method	DN AbstraGrid sparse								
	ResNet-18	L2D	ACVC	Crafting	SelfReg	MIRO	SIMPLE	ERM++	Average
No blur	2.6	3.0	6.3	6.9	3.1	2.9	2.5	2.1	3.7
MEMO	2.6	2.8	6.3	6.9	2.9	2.9	2.5	2.0	3.6
TENT 1%	1.5	1.6	1.6	1.3	1.2	0.9	0.9	0.9	1.2
†TENT 10%	1.2	1.3	1.8	1.1	1.4	1.4	1.3	1.3	1.4
†TENT 20%	1.2	1.5	3.0	1.3	1.6	2.8	2.2	2.1	2.0
†TENT 30%	1.4	1.8	5.6	1.5	2.4	3.9	3.2	3.2	2.9
†TENT 40%	2.1	2.4	9.9	1.6	2.9	5.6	4.7	4.1	4.2
†TENT 50%	2.5	3.2	13.3	1.9	3.1	6.4	5.8	**4.7**	5.1
Squint! (ours)	**4.1**	**5.0**	**14.4**	**10.9**	**4.7**	**7.5**	**7.5**	**4.7**	**7.3**

Table 2: **Test-time adaptation.** MEMO and TENT. Dagger (†) denotes oracle batches with 10%–50% sparse samples. Squint! achieves the highest accuracy without batch assumptions.

Metric	DN AbstraGrid Sparse (To improve)				Original Domain (Control dataset)			
	Negatives		Positives		Negatives		Positives	
	No blur	Squint!	No blur	Squint!	No blur	Squint!	No blur	Squint!
Acc. (%)	98.6	99.2↑	11.3	16.6↑	100	100✓	100	100✓

Table 3: **Sparse obfuscation in content moderation.** Squint! improves recognition on sparse transformations (↑) while preserving performance on the original images (✓).

a realistic setting in which sparse inputs are rare, while 10%–50% provide increasingly favorable oracle conditions. At 1% sparsity, TENT reduces average sparse accuracy from 3.7% to 1.2%. In fact, TENT remains below the no blur baseline when sparse images constitute up to 30% of the batch, and only begins to improve once this fraction reaches 40%. Even with an oracle mixture containing 50% sparse images, TENT reaches only 5.1%, compared with 7.3% for Squint!. MEMO also fails to improve the original models. Thus, under realistic sparse frequencies, MEMO provides no benefit while TENT harms sparse recognition. The results are summarized in Table 2.

Content moderation. Finally, we evaluate Squint! for content moderation using GPT 5.4 as a binary hate-symbol detector. We use the same fixed prompt for every image, asking whether a recognizable hate symbol is visibly present and requiring either *hate* or *not hate*. Starting from the publicly available Hate Symbols dataset [2], we retain 359 images that GPT 5.4 correctly recognizes as containing a hate symbol. We then apply the least abstract versions of our four sparse transformations and manually retain only images in which the symbol remains recognizable, yielding 213–314 images per sparse domain. These form realistic sparse moderation attacks, as hateful images detected in their original form become substantially harder for the same filter to recognize after sparse transformation while remaining human-recognizable. We also construct a negative set from the Mini DomainNet photo domain using images classified by GPT 5.4 as *not hate*, and apply the same transformations. We query GPT 5.4 before and after applying the blur selected by Squint!. As shown in Table 3, Squint! increases accuracy on sparse positives from 11.3% to 16.6% and on sparse negatives from 98.6% to 99.2%. Accuracy on the corresponding original positives and negatives remains at 100% before and after Squint!, showing that the gain on sparse moderation attacks does not disrupt recognition of the original inputs. The complete prompt and dataset construction protocol are provided in Appendix A.3.

7 Conclusion

We studied the recognition-performance degradation caused by sparsifying the visual evidence that carries relevant information. Our experiments show that such practices can be used to obfuscate harmful image content in a way that is still recognized by humans but can pass automated content filters. Unlike adversarial attacks that are usually tailored to a specific and known network, sparse-domain images are problematic for a wide range of architectures, including modern LMMs. In a controlled human study, we showed that blur improves recognition for both humans and models, but benefits models disproportionately. We demonstrated the practical relevance of this gap in content moderation, where sparse transformations make hate symbols substantially harder for automated filters to recognize, yet still human-recognizable. To study this regime systematically, we introduced DN AbstraGrid, which separates sparse target domains from non-sparse and original control domains. We proposed Squint!, a lightweight, model-agnostic module that predicts when and how much to blur from a compact radial Fourier power-spectrum descriptor. Trained without access to the target sparse domains, a single Squint! model transfers across 12 vision-language models and 7 domain-generalization methods, improving sparse-domain recognition while preserving control-domain performance.

[2] https://github.com/OpnBrdrsAdvct/hate-symbols

AI USE STATEMENT

Generative AI tools were used to assist with drafting and polishing parts of the manuscript, retrieval and discovery of relevant literature, and coding support such as suggesting implementations and helping with debugging. AI tools were not used to independently conduct the research: the research questions, experimental design, dataset construction, execution and validation of experiments, analysis of results, and scientific conclusions were carried out by the authors. All AI-assisted text, code, and literature suggestions were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

ETHICS STATEMENT

Our human study involved 16 adult volunteers performing a non-invasive visual classification task. Participation was voluntary, participants were informed about the nature of the task, and no sensitive personal data or information allowing individual participants to be identified was collected or reported. The study was conducted in accordance with the ethical principles and standards applicable at the authors' institution. A supporting institutional ethics statement has been obtained; identifying institutional information is omitted from the submission to preserve double-blind review.

Our content-moderation experiment uses publicly available hate-symbol imagery to study a failure mode in which sparse transformations can reduce automated recognition while remaining interpretable to humans. Such transformations could potentially be misused to evade moderation systems; our purpose in studying this vulnerability is to characterize it and evaluate Squint! as a defensive mechanism for improving recognition of these inputs.

REPRODUCIBILITY STATEMENT

We provide the experimental and implementation details required to reproduce the work throughout the paper and Appendix. The construction of DN AbstraGrid, including the exact rendering rules and four abstraction levels for each domain, is described in section 3 and Appendix A.1. The Squint! training procedure, Fourier descriptor, blur candidate set, weak-target construction, architecture, objective, model selection, and inference procedure are specified in Figure 4, with optimization and fine-tuning details in Appendix A.4. The human-study setup, stimuli, instructions, evaluation protocol, and statistical test are given in section 4 and Appendix A.2. The content-moderation experiment, including the exact GPT 5.4 prompt, positive and negative set construction, sparse transformations, and evaluation protocol, is described in section 6 and Appendix A.3. Additional ablations and computational analysis are reported in Appendix A.5, full per-domain results in Appendix A.6, and examples covering all DN AbstraGrid domains and abstraction levels in Appendix A.7. Code for Squint!, DN AbstraGrid generation, and the evaluation pipelines will be released upon acceptance.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 2012.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Nicholas Baker and Philip J. Kellman. Shape from dots: A window into abstraction processes in visual perception. *Frontiers in Computer Science*, 2024.

Moshe Bar, Karim S. Kassam, Avniel S. Ghuman, Jasmine Boshyan, Annette M. Schmid, Andrew M. Schmidt, Anders M. Dale, Matti S. Hämäläinen, Ksenija Marinkovic, Daniel L. Schacter, Bruce R. Rosen, and Eric Halgren. Top-down facilitation of visual recognition. *Proceedings of the National Academy of Sciences*, 2006.

Irving Biederman and Eric E. Cooper. Size invariance in visual object priming. *Journal of Experimental Psychology: Human Perception and Performance*, 1992.

Irving Biederman and Ginny Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 1988.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Abdul Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025.

Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 1983.

John Canny. A computational approach to edge detection. *PAMI*, 1986.

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *ECCV*, 2022.

Seokeon Choi, Debasmit Das, Sungha Choi, Seunghan Yang, Hyunsin Park, and Sungrack Yun. Progressive random convolutions for single domain generalization. In *CVPR*, 2023.

Ilke Cugu, Massimiliano Mancini, Yanbei Chen, and Zeynep Akata. Attention consistency on visual corruptions for single-source domain generalization. In *CVPRW*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 1973.

Nikos Efthymiadis, Giorgos Tolias, and Ondrej Chum. Edge augmentation for large-scale sketch recognition without sketches. In *ICPR*, 2022.

Nikos Efthymiadis, Giorgos Tolias, and Ondřej Chum. Crafting distribution shifts for validation and training in single source domain generalization. In *WACV*, 2025.

David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 1987.

Anat R. Fintzi and Bradford Z. Mahon. A bimodal tuning curve for spatial frequency across left and right human orbital frontal cortex during object recognition. *Cerebral Cortex*, 2014.

Guy Gaziv, Michael J. Lee, and James J. DiCarlo. Strong and precise modulation of human percepts via robustified anns. In *NeurIPS*, 2023.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

Sivan Harary, Eli Schwartz, Assaf Arbelle, Peter Staar, Shady Abu-Hussein, Elad Amrani, Roei Herzig, Amit Alfassy, Raja Giryes, Hilde Kuehne, et al. Unsupervised domain generalization by learning a bridge across domains. In *CVPR*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

Hojin Jang and Frank Tong. Improved modeling of human vision by incorporating robustness to blur in convolutional neural networks. *Nature Communications*, 2024.

Daehee Kim, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ICCV*, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

Haotian Li and Jianbo Jiao. An explorative study on abstract images and visual representations learned from them. In *BMVC*, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023a.

Ziyue Li, Kan Ren, Xinyang Jiang, Yifei Shen, Haipeng Zhang, and Dongsheng Li. SIMPLE: Specialized model-sample matching for domain generalization. In *ICLR*, 2023b.

Tony Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings of the CERN School of Computing*, 1996.

Eftichia Mavridaki and Vasileios Mezaris. No-reference blur assessment in natural images using fourier transform and spatial pyramids. In *ICIP*, 2014.

Aude Oliva, Antonio Torralba, Anne Guerin-Dugue, and Jeanny Herault. Global semantic classification of scenes using power spectrum templates. In *Proceedings of the Challenge of Image Retrieval*, 1999.

Sven Panis, Joeri De Winter, Joachim Vandekerckhove, and Johan Wagemans. Identification of everyday objects on the basis of fragmented outline versions. *Perception*, 2008.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

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 *ICML*, 2021.

Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1972.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Elsa Scialom, Udo A. Ernst, David Rotermund, and Michael H. Herzog. Object recognition from sparse simulated phosphenes and curved segments. *Vision Research*, 2026.

Johannes J. D. Singer, Katja Seeliger, Tim C. Kietzmann, and Martin N. Hebart. From photos to sketches: How humans and deep neural networks process objects across different levels of visual abstraction. *Journal of Vision*, 2022.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

Florian Stimberg, Ayan Chakrabarti, Chun-Ta Lu, Hussein Hazimeh, Otilia Stretcu, Wei Qiao, Yintao Liu, Merve Kaya, Cyrus Rashtchian, Ariel Fuxman, Mehmet Tek, and Sven Gowal. Benchmarking robustness to adversarial image obfuscations. In *NeurIPS*, 2023.

Ajay Subramanian, Elena Sizikova, Najib J. Majaj, and Denis G. Pelli. Spatial-frequency channels, shape bias, and adversarial robustness. In *NeurIPS*, 2023.

Piotr Teterwak, Kuniaki Saito, Theodoros Tsiligkaridis, Kate Saenko, and Bryan A. Plummer. ERM++: An improved baseline for domain generalization. In *WACV*, 2025.

John Theios and Paul C. Amrhein. Theoretical analysis of the cognitive processing of lexical and pictorial stimuli: Reading, naming, and visual and conceptual comparisons. *Psychological Review*, 1989.

Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 2003.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. In *arXiv*, 2025.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021a.

Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021b.

Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2 edition, 2004.

M. Wolff, T. Klein, E. Rusak, F.A. Wichmann, and W. Brendel. Low-pass filtering improves behavioral alignment of vision models. In *ICLR*, 2026.

Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *IJCV*, 2017.

Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *ICLR*, 2021.

Sou Yoshihara, Taiki Fukiage, and Shin'ya Nishida. Does training with blurred images bring convolutional neural networks closer to humans with respect to robust object recognition and internal representations? *Frontiers in Psychology*, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022.

A APPENDIX

A.1 DN ABSTRAGRID DOMAIN DETAILS

Let x denote an image, D an abstraction family, and K a domain-specific control parameter. We define an abstraction $a_{D,K}(x)$ as a re-rendering of x using the primitives and rendering rules associated with D, with K controlling the abstraction level. Depending on the family, K may determine the number of visual primitives, the resolution of a rendering grid, or the strength of a structural approximation. Varying K produces a discrete, ordered set of abstraction levels within each family.

DN AbstraGrid contains eight abstraction families, each instantiated at four ordered abstraction levels. The four sparse families form the target partition used throughout the paper, while the four non-sparse families form a control partition. The original Mini DomainNet domains provide an additional control with the same semantic classes.

For each abstraction family D, the parameter K determines the abstraction level. The interpretation of K is family-specific and is described below.

Sparse Target Domains. The sparse partition consists of four abstraction families whose visual support is distributed across disconnected components.

ASCII. The ASCII domain renders the image using character tokens placed on a regular grid. Each cell is mapped to a character according to local brightness and colored using the average RGB value of the corresponding image region. Because the characters form separated visual elements, the resulting depiction is sparse. Here, K controls the grid resolution through the number of characters along the smaller image dimension, with

$$K \in \{20, 25, 30, 35\}.$$

Halftone. The Halftone domain renders the image as a regular grid of colored dots whose diameters vary with local intensity. The spatial separation between neighboring dots produces a sparse depiction. Here, K controls the dot-grid resolution through the number of dots along the smaller image dimension, with

$$K \in \{20, 25, 30, 35\}.$$

H-Lines. The H-Lines domain renders the image using horizontal rows placed at regularly spaced vertical positions. Within each row, local brightness determines the thickness of the rendered marks. The gaps between neighboring rows produce a sparse representation. Here, K controls the number of horizontal rows, with

$$K \in \{20, 25, 30, 35\}.$$

Cardiogram. The Cardiogram domain renders the image using vertically anchored line trajectories. A fixed set of columns is sampled, and each trajectory evolves from top to bottom with horizontal displacement determined by local brightness. Each line has a fixed thickness of two pixels, and at each position the color assigned to these two pixels is the average color of the corresponding two pixels in the original image. The separated trajectories produce a sparse representation. Here, K controls the number of line trajectories, with

$$K \in \{25, 35, 45, 55\}.$$

Non-Sparse Control Domains. The non-sparse partition consists of four abstraction families whose visual support is not distributed across disconnected components.

Hexagons. The Hexagons domain replaces the image with a hexagonal tessellation, filling each hexagon with the average color of the pixels it covers. Neighboring hexagons tile the image plane without gaps, making the representation non-sparse. Here, K specifies the number of hexagons along the smaller image dimension, while the larger dimension follows the image aspect ratio. We use

$$K \in \{20, 25, 30, 35\}.$$

Tiles. The Tiles domain renders the image as a square mosaic. The image is divided into a regular grid, and each tile is filled with the average color of the corresponding image region. Because the

tiles cover the image plane without gaps, the representation is non-sparse. Here, K controls the number of tiles along the smaller image dimension, with

$$K \in \{20, 25, 30, 35\}.$$

Polygonal Approximation. The Polygonal Approximation domain Ramer (1972); Douglas & Peucker (1973) is based on edge extraction followed by contour simplification. We detect edges, remove small connected components, extract contours, and approximate each contour using a polygonal chain. Here, K controls the contour-simplification tolerance relative to contour perimeter, with

$$K \in \{0.1, 0.05, 0.01, 0.005\},$$

ordered from coarser to finer approximation.

SuperPixel. The SuperPixel domain partitions the image into SLIC regions Achanta et al. (2012) and replaces each region with its average color. Here, K controls the target number of SuperPixels, with

$$K \in \{20, 40, 60, 80\}.$$

Mini DomainNet Control Domains. In addition to the eight abstraction families, DN AbstraGrid includes the original Mini DomainNet domains. These share the same semantic classes as the abstraction partitions and serve as an additional control for evaluating whether improvements on sparse domains come at the expense of performance on standard visual domains.

A.2 Human Study

Participants and experimental setup. We conducted the study in person with 16 adult participants. All participants were tested under the same experimental setup, using the same monitor and a fixed viewing distance of approximately 1 m. Following prior work on stimulus size in object recognition, we selected the presentation size to lie within the approximately 4–6° range associated with efficient object identification Biederman & Cooper (1992); Ware (2004); smaller pictorial stimuli have also been shown to incur additional processing costs Theios & Amrhein (1989). Because our stimuli have varying aspect ratios, we deliberately chose the upper end of this range, setting the longest image dimension to approximately 6° of visual angle. This choice provides additional margin for the shorter dimension to remain above the lower end of the recommended range, while remaining consistent with recent controlled human object-recognition experiments that also use approximately 6° stimuli Gaziv et al. (2023). At a distance of 1 m, 6° corresponds to approximately 10.5 cm.

Stimuli and protocol. The source images were drawn from the 16 ImageNet categories used by Geirhos et al. (2019). From these, we construct a cleaned human-study set comprising 512 rendered stimuli in total, spanning the four sparse DN AbstraGrid families and their four abstraction levels. This corresponds to 32 stimuli for each family–level combination. The same 512 stimuli are used for both the human and model evaluations. Each participant classified 32 images, and no participant was shown more than one rendering originating from the same source image.

Before starting the experiment, every participant received the same instructions:

> The task is to classify these abstract images into one of the following 16 classes. Please take some time to read the classes carefully. You may not squint your eyes during this test. For each image, after your initial classification you will have a second round in which you will have control over different blur strengths on the image. Choose the blur that maximizes your confidence in what the image depicts and then classify again. You may choose the same or a different class in this round. You will classify a total of 32 images.

For each image, participants first classified the unblurred sparse rendering. They were then given control over the same nine Gaussian blur levels used throughout the paper and selected the blur level at which they felt most confident before classifying the image again. The corresponding model evaluation follows the same two-stage structure, except that the blur level is selected by minimum prediction entropy. Figure 6 reports the human accuracy before and after participant-selected blur together with the median selected blur strength at each abstraction level. Blur improves accuracy across all four sparse families, while stronger blur is generally selected for the more strongly abstracted renderings.

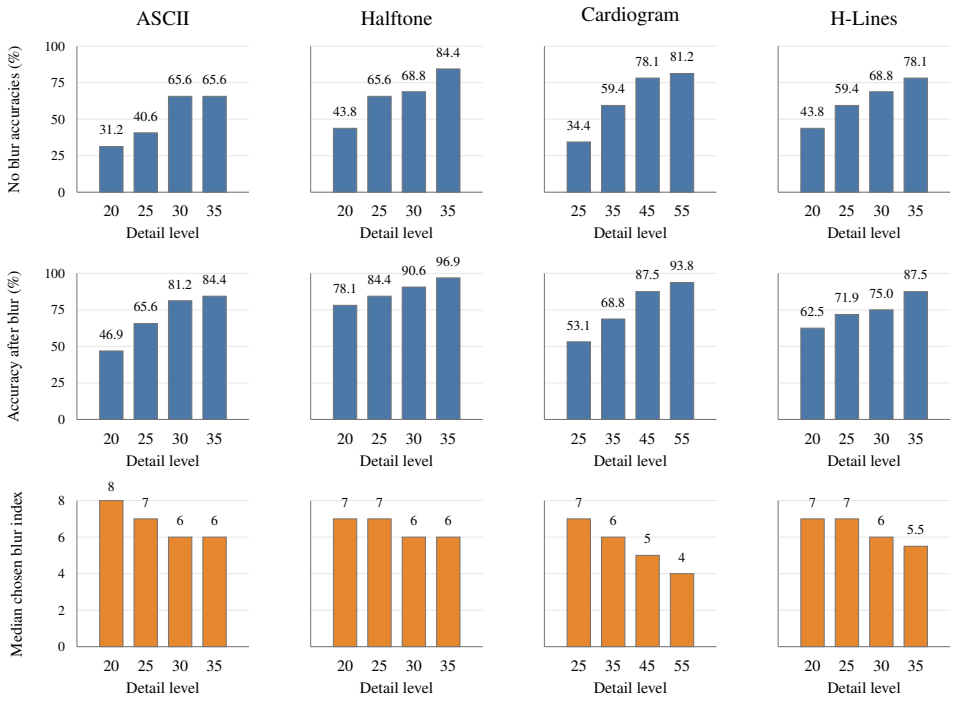

Figure 6: Human-study accuracies without extra blur and after the participant-selected blur.

Statistical significance. We evaluate statistical significance at the participant level. For each participant, we compute classification accuracy before and after participant-selected blur over their 32 trials and apply an exact one-sided sign test to the paired accuracy differences. Across all participants, accuracy increases from 60.5% to 76.8%, a gain of 16.3 percentage points. All 16 participants achieve higher accuracy after blur, with no ties or decreases, giving $p = 1.53 \times 10^{-5}$.

Human–model comparison. Table 4 provides a detailed comparison between humans and the evaluated recognition models on the same 512 stimuli. In addition to standard accuracy and the sparsity-vulnerability ratio, we report *human-target accuracy*. For this metric, the participant's initial classification is treated as the target for each image rather than the ImageNet ground-truth label. Thus, if an image belongs to class 1 but the participant classifies it as class 2, a model prediction of class 2 is counted as correct. This measures whether models interpret sparse images in the same way as humans, including cases in which the human interpretation differs from the dataset label.

We also compare the amount of blur preferred by humans and models. For each abstraction level, we compute the absolute difference between the median human-selected blur index and the median blur index selected by the model using MC-E, normalized by the maximum possible difference of eight. The resulting normalized ℓ_1 distance therefore lies in $[0, 1]$, with lower values indicating closer agreement with human blur preference. A value of 0.125, for example, corresponds to a difference of one blur level. Domain-level values are averaged over abstraction levels, and the overall value is averaged across the four sparse domains.

A.3 Content Moderation Evaluation

Moderation task. We use GPT 5.4 as a binary hate-symbol detector. Every image is evaluated independently using the same fixed prompt. The task is deliberately defined as presence detection rather than an assessment of the intent or context in which a symbol appears. A recognizable hate symbol should be detected, whether it appears historically, educationally, critically, satirically, or supportively. GPT 5.4 is required to return exactly *hate* or *not hate*.

The exact prompt used for every image is

Model	ASCII					Halftone					Cardiogram				
	ACC (%) ↑ no blur	ACC (%) ↑ MC-E	Sparsity vulnerability ↓	human target Acc (%) ↑	blur level human L1 ↓	ACC (%) ↑ no blur	ACC (%) ↑ MC-E	Sparsity vulnerability ↓	human target Acc (%) ↑	blur level human L1 ↓	ACC (%) ↑ no blur	ACC (%) ↑ MC-E	Sparsity vulnerability ↓	human target Acc (%) ↑	blur level human L1 ↓
Human	50.8	69.5	1.368	100.0	0.000	65.6	87.5	1.334	100.0	0.000	63.3	75.8	1.197	100.0	0.000
ResNet18	11.7	14.8	1.266	15.6	0.500	13.3	18.8	1.412	13.3	0.469	10.9	25.0	2.285	14.8	0.266
ACVC	6.2	32.0	5.125	14.1	0.125	5.5	37.5	6.856	5.5	0.062	18.8	31.2	1.667	18.0	0.391
Crafting	14.1	34.4	2.445	10.9	0.422	8.6	45.3	5.275	3.9	0.188	31.2	49.2	1.575	25.8	0.203
L2D	12.5	10.9	0.875	11.7	0.594	11.7	12.5	1.067	10.2	0.344	14.8	16.4	1.106	15.6	0.203
SelfReg	11.7	17.2	1.467	9.4	0.078	11.7	16.4	1.400	12.5	0.047	14.1	18.8	1.334	14.8	0.156
Simple	8.6	31.2	3.638	7.0	0.266	15.6	33.6	2.150	11.7	0.188	20.3	36.7	1.808	21.9	0.109
ERM++	8.6	15.6	1.818	8.6	0.312	8.6	25.8	3.001	7.0	0.031	14.8	27.3	1.842	14.8	0.375
MIRO	11.7	31.2	2.666	11.7	0.156	18.0	39.8	2.217	13.3	0.109	16.4	34.4	2.095	18.8	0.219
CLIP-ViT-B-16	7.8	45.3	5.802	5.5	0.547	25.8	75.0	2.909	21.9	0.031	34.4	67.2	1.954	28.1	0.062
+ fine-tuned blur	9.4	49.2	5.247	6.2	0.594	25.8	82.8	3.212	21.9	0.000	35.9	69.5	1.935	30.5	0.031
+ fine-tuned imgaug	10.9	52.3	4.784	8.6	0.547	30.5	86.7	2.846	27.3	0.000	46.1	73.4	1.593	39.8	0.031
+ fine-tuned full	11.7	53.1	4.532	9.4	0.516	32.8	86.7	2.643	28.1	0.000	43.0	76.6	1.782	39.1	0.016
CLIP-ViT-L-14	12.5	56.2	4.500	9.4	0.438	35.2	85.2	2.422	35.2	0.062	50.0	79.7	1.594	43.0	0.062
CLIP-ViT-H-14	14.1	72.7	5.168	10.9	0.188	43.8	90.6	2.071	40.6	0.031	53.9	80.5	1.493	45.3	0.031
SigLIP-ViT-B-16	7.8	39.8	5.101	5.5	0.641	14.1	72.7	5.168	13.3	0.031	45.3	66.4	1.466	37.5	0.062
SigLIP-ViT-L-16-384	15.6	58.6	3.751	15.6	0.172	27.3	81.2	2.972	28.1	0.125	43.8	78.1	1.786	39.1	0.172
SigLIP-ViT-SO400M-14-384	10.2	67.2	6.613	9.4	0.297	36.7	87.5	2.383	35.9	0.125	45.3	78.1	1.724	37.5	0.094
SigLIP2-ViT-B-16-256	13.3	50.0	3.765	10.9	0.422	28.9	78.9	2.730	25.8	0.031	44.5	76.6	1.719	37.5	0.109
SigLIP2-ViT-L-16-256	28.9	78.1	2.702	26.6	0.094	48.4	90.6	1.871	43.0	0.078	58.6	86.7	1.480	47.7	0.125
SigLIP2-ViT-SO400M-16-384	21.1	79.7	3.779	16.4	0.156	53.1	91.4	1.721	46.9	0.156	57.8	85.9	1.487	50.8	0.188
PE-Core-B-16	15.6	76.6	4.901	13.3	0.125	53.1	90.6	1.706	49.2	0.016	50.0	80.5	1.609	38.3	0.031
PE-Core-L-14-336	31.2	88.3	2.825	26.6	0.047	85.2	96.1	1.128	65.6	0.062	77.3	88.3	1.141	59.4	0.000
PE-Core-bigG-14-448	55.5	90.6	1.634	39.8	0.094	95.3	97.7	1.025	64.8	0.094	78.9	92.2	1.168	60.2	0.141

Model	H-Lines					Average				
	ACC (%) ↑ no blur	ACC (%) ↑ MC-E	Sparsity vulnerability ↓	human target Acc (%) ↑	blur level human L1 ↓	ACC (%) ↑ no blur	ACC (%) ↑ MC-E	Sparsity vulnerability ↓	human target Acc (%) ↑	blur level human L1 ↓
Human	62.5	74.2	1.187	100.0	0.000	60.5	76.8	1.269	100.0	0.000
ResNet18	15.6	18.8	1.200	12.5	0.188	12.9	19.3	1.500	**14.1**	0.355
ACVC	13.3	31.2	2.353	11.7	0.188	10.9	33.0	3.017	12.3	0.191
Crafting	16.4	25.0	1.523	11.7	0.281	**17.6**	**38.5**	2.189	13.1	0.273
L2D	13.3	17.2	1.294	12.5	0.062	13.1	14.3	**1.089**	12.5	0.301
SelfReg	11.7	17.2	1.467	10.2	0.062	12.3	17.4	1.413	11.7	**0.086**
Simple	11.7	27.3	2.333	7.8	0.094	14.1	32.2	2.592	12.1	0.164
ERM++	11.7	27.3	2.333	7.0	0.094	10.9	24.0	2.196	9.4	0.203
MIRO	10.2	31.2	3.076	10.2	0.031	14.1	34.2	2.431	13.5	0.129
CLIP-ViT-B-16	18.8	40.6	2.166	18.8	0.078	21.7	57.0	2.631	18.6	0.180
+ fine-tuned blur	18.8	44.5	2.375	18.8	0.047	22.5	61.5	2.739	19.3	0.168
+ fine-tuned imgaug	23.4	50.8	2.166	25.8	0.016	27.7	65.8	2.374	25.4	0.148
+ fine-tuned full	25.0	47.7	1.906	25.8	0.016	28.1	66.0	2.348	25.6	0.137
CLIP-ViT-L-14	19.5	51.6	2.640	18.0	0.094	29.3	68.2	2.326	26.4	0.164
CLIP-ViT-H-14	23.4	57.0	2.433	22.7	0.016	33.8	75.2	2.226	29.9	0.066
SigLIP-ViT-B-16	15.6	43.8	2.801	16.4	0.094	20.7	55.7	2.689	18.2	0.207
SigLIP-ViT-L-16-384	16.4	52.3	3.190	16.4	0.141	25.8	67.6	2.621	24.8	0.152
SigLIP-ViT-SO400M-14-384	21.1	57.0	2.704	13.3	0.172	28.3	72.5	2.559	24.0	0.172
SigLIP2-ViT-B-16-256	12.5	46.1	3.687	10.9	0.266	24.8	62.9	2.536	21.3	0.207
SigLIP2-ViT-L-16-256	19.5	63.3	3.240	18.8	0.062	38.9	79.7	2.050	34.0	0.090
SigLIP2-ViT-SO400M-16-384	28.9	63.3	2.189	24.2	0.109	40.2	80.1	1.991	34.6	0.152
PE-Core-B-16	21.9	72.7	3.321	21.9	0.016	35.2	80.1	2.278	30.7	**0.047**
PE-Core-L-14-336	40.6	75.0	1.846	32.0	0.109	58.6	86.9	1.483	45.9	0.055
PE-Core-bigG-14-448	43.8	82.8	1.893	35.2	0.125	**68.4**	**90.8**	**1.329**	**50.0**	0.113

Table 4: **Detailed human–model comparison on the human-study subset.** For each sparse domain, we report accuracy without blur, accuracy after MC-E blur selection, and the resulting sparsity-vulnerability ratio. *Human target Acc.* measures agreement with the participant's predicted class rather than the dataset ground-truth label. *Human L1* is the normalized absolute distance between the median human-selected and model-selected blur indices, averaged over abstraction levels; lower values indicate closer agreement with human blur preference. The final columns report averages across the four sparse domains.

> Determine whether the image visibly contains a hate symbol.
>
> A hate symbol means a recognizable emblem, insignia, rune, flag, gesture, icon, number combination, or other visual symbol associated with Nazism, white supremacy, antisemitism, racism, or an extremist hate organization.
>
> This is presence detection only. Answer "hate" whenever a recognizable hate symbol is visible, regardless of whether the image uses it historically, educationally, critically, satirically, or supportively.
>
> Do not classify an ordinary religious symbol, national flag, geometric shape, letter, or number as hate unless it is clearly presented as a recognized hate-symbol variant.
>
> Reply with exactly one of these two lowercase answers and nothing else:
>
> hate
>
> not hate

Positive set construction. We begin with all 1,215 images from the publicly available Hate Symbols dataset mentioned in the main paper. We do not independently define which examples in the dataset should count as hate symbols for this experiment. Instead, we query GPT 5.4 using the prompt above and retain only the 359 images for which it responds *hate*. Consequently, every retained original image is correctly detected as positive by the moderation filter before any sparse transformation is applied.

We then transform each retained image using the least abstract version of each of the four sparse DN AbstraGrid families. Specifically, we use ASCII 35, Halftone 35, H-Lines 35, and Cardiogram 55. The goal is to construct examples that are harder for the moderation filter while still preserving human recognizability of the hate symbol. We therefore manually inspect the transformed images and retain only those in which a human observer can still recognize the hate symbol. After this filtering, the resulting set contains 299 ASCII images, 285 Halftone images, 213 H-Lines images, and 314 Cardiogram images.

These retained transformed images constitute successful sparse obfuscations for the moderation setting. Their original versions are detected by GPT 5.4, while the sparse renderings remain recognizable to humans but become substantially harder for the same automated filter to identify.

Negative set construction. We additionally construct a negative set from the photo domain of Mini DomainNet. We select 268 images that GPT 5.4 classifies as *not hate* using the same moderation prompt. We apply the same four sparse transformations and abstraction levels used for the positive set. This negative set allows us to test whether Squint! improves recognition of sparse hate symbols without causing ordinary sparse images to be incorrectly classified as hate symbols.

Evaluation protocol. For every sparse image, we first query GPT 5.4 on the unblurred version using the fixed moderation prompt. We then apply the blur level predicted by Squint! and query GPT 5.4 again using exactly the same prompt. For the positive set, a prediction of *hate* is counted as correct. For the negative set, a prediction of *not hate* is counted as correct.

We perform the same before and after Squint! evaluation on the corresponding original images. Since the positive originals were selected from images initially classified as *hate*, and the negative originals were selected from images initially classified as *not hate*, their no blur accuracy is 100% by construction. This original domain evaluation serves as a control for whether the blur selected by Squint! disrupts moderation decisions on images that are already handled correctly.

Results. Table 3 reports the results. On the sparse positive set, GPT 5.4 recognizes only 11.3% of the human recognizable hate symbol images without blur. Applying the blur selected by Squint! increases this to 16.6%. On the sparse negative set, accuracy also increases from 98.6% to 99.2%, indicating that the gain on positive examples does not come from indiscriminately increasing hate symbol predictions. On the corresponding original images, accuracy remains at 100% for both the positive and negative sets after applying Squint!. These results show that Squint! partially recovers recognition of sparse moderation attacks while preserving the behavior of the moderation filter on the corresponding original inputs.

A.4 Implementation Details

Domain generalization training setup. For the domain generalization experiments, MIRO, SIMPLE, and ERM++ use a ResNet50 backbone and follow a multi-source domain generalization setting. They are trained on the training splits of all Mini DomainNet domains. Consequently, their high accuracy on Mini DomainNet in Table 1 reflects an in domain evaluation rather than generalization to an unseen domain. In contrast, L2D, ACVC, Crafting, and SelfReg are trained in a single-source domain generalization setting using only the photo domain of the Mini DomainNet training split. ResNet18 is included as the corresponding baseline for this single-source setting.

Training. We train Squint! with a cross-entropy loss L_{CE} and a diversity regularizer L_E on its outputs, where L_E maximizes the entropy of the batch-averaged prediction. The total objective is

$$L_{\text{total}} = L_{\mathrm{CE}} - \lambda L_E,$$

with $\lambda = 10$. We optimize with stochastic gradient descent using a learning rate of 3×10^{-2}, weight decay of 10^{-4}, batch size 256, for one pass over all 250,000 images. Squint! is a residual MLP with layer normalization, a 64-to-64 hidden layer, ReLU, dropout 0.1, and a 64-to-9 output layer followed by layer normalization, dropout 0.1, and softmax, for 5,019 trainable parameters. Owing to the small size of Squint! and its low-dimensional input and output, each run takes only a few seconds on an NVIDIA Tesla A100 GPU.

CLIP fine-tuning. We fine-tune CLIP-B on a cluster with 4 GPUs using a total of 42 million samples. The per-GPU batch size is 256 image–caption pairs, and gradients are accumulated over 32 forward passes before each parameter update. The learning rate is set to 5×10^{-6}. Following fine-tuning, we perform weight averaging between the pretrained and fine-tuned checkpoints, assigning weights 0.6 and 0.4, respectively. The final model θ is given by

$$\theta = 0.6\,\theta_{PT} + 0.4\,\theta_{FT}.$$

A.5 More Experiments.

Why Gaussian Blur. We compare Gaussian blur with downsampling and ideal low-pass filtering in Figure 7. All three can improve recognition on DN AbstraGrid Sparse, but differ substantially in how

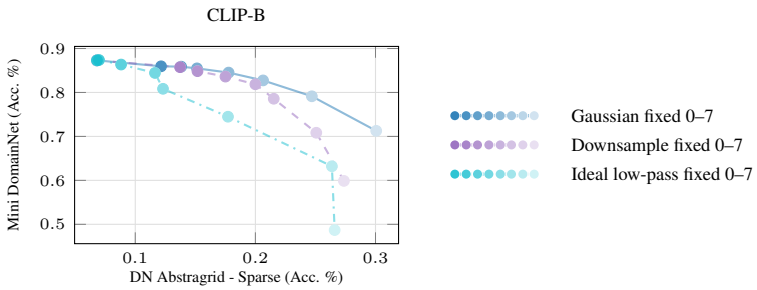

Figure 7: **Trade-off for different blurs.** Gaussian blur provides the best trade-off between sparse-domain gains and preserving Mini DomainNet accuracy.

quickly they degrade performance on Mini DomainNet as their strength increases. Gaussian blur provides the most favorable trade-off, reaching higher sparse accuracy while retaining substantially more of the original-domain accuracy. We therefore use Gaussian blur throughout our experiments.

Why 9 Blurs. We study the number of blur levels used by MC-E in Figure 8. Increasing the candidate set from 1 to 9 produces large gains on sparse domains for both Crafting and CLIP-B. Beyond 9 levels, performance begins to saturate. Increasing from 9 to 17 blurs improves MC-E only from 20.1% to 20.6% for Crafting and from 29.4% to 30.5% for CLIP-B, despite nearly doubling the required number of forward passes. We therefore use 9 blur levels as a practical accuracy–computation trade-off.

Why MC-E. In Figure 9, we compare probability voting, selecting the prediction with the maximum class probability (MC-P), and selecting the blur with the lowest prediction entropy (MC-E). MC-E performs best across all three evaluation sets. On DN AbstraGrid Sparse, average accuracy increases from 28.7% with voting or MC-P to 29.4% with MC-E. MC-E also improves the non-sparse control from 52.2% to 52.8% and Mini DomainNet from 86.2% to 86.6%. We therefore use MC-E for multi-blur inference.

Combining MC-E with Squint! further increases sparse accuracy from 29.4% to 29.9% while preserving the control-domain results. Squint! additionally prunes blur levels weaker than its predicted level, reducing the number of recognition forward passes required by MC-E.

Latency. The accuracy–computation trade-off is shown in Figure 10. Single-pass Squint! adds less than 0.01 GFLOPs to each backbone and therefore has negligible computational overhead. MC-E instead evaluates all nine blur levels, increasing CLIP-B from 35.1 to 316.4 GFLOPs per image and CLIP-H from 334.7 to 3012.3 GFLOPs per image. On sparse inputs, Squint!-guided pruning reduces MC-E computation to 261.9 GFLOPs for CLIP-B and 2493.0 GFLOPs for CLIP-H while also improving accuracy. The computational advantage of pruning, therefore, becomes increasingly important as the recognition backbone grows.

A.6 Full Results

In Figure 11, we show the results of Squint! for every detail level on the full DN AbstraGrid, for ACVC, Crafting, CLIP-B, CLIP-B Full Ft, MIRO, and SIMPLE as backbones.

Full result tables for Table 1 of the main paper are provided in Table 5, Table 6, and Table 7.

A.7 DN AbstraGrid Overview

In Figure 12, and Figure 13, we show example images from DN AbstraGrid across all abstraction domains (columns) and all detail levels (rows).

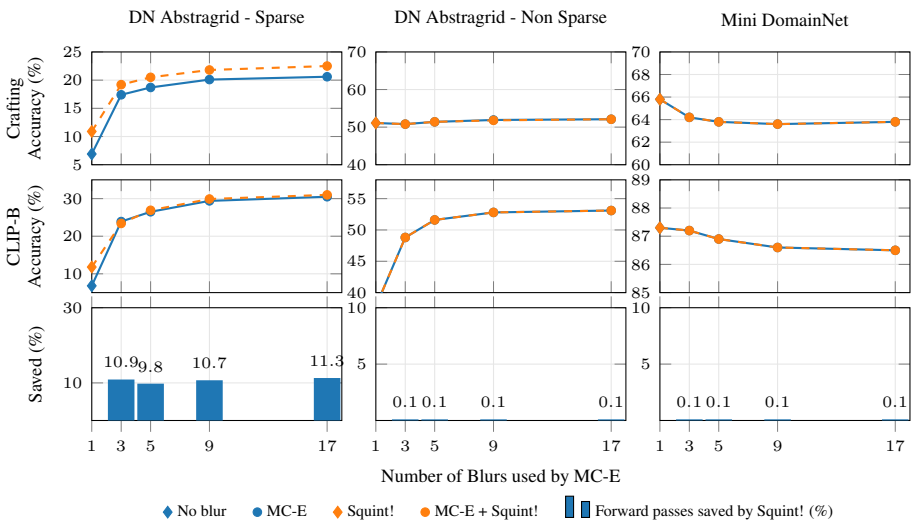

Figure 8: **Number of MC-E blur levels.** Performance saturates at 9 blur levels, motivating our use of 9 candidates for MC-E.

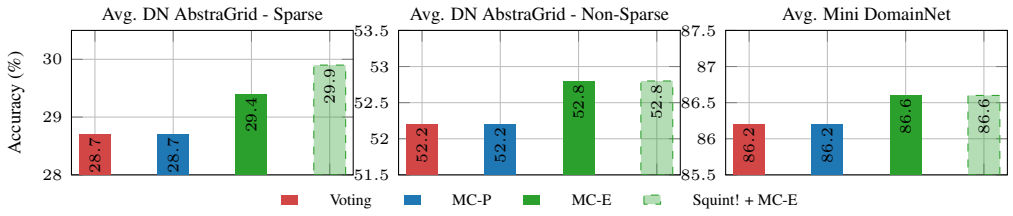

Figure 9: **Multi-blur strategies**. MC-E performs better on both Mini DomainNet and sparse DN AbstraGrid. Ensemble pruning guided by Squint! improves further the sparse domains.

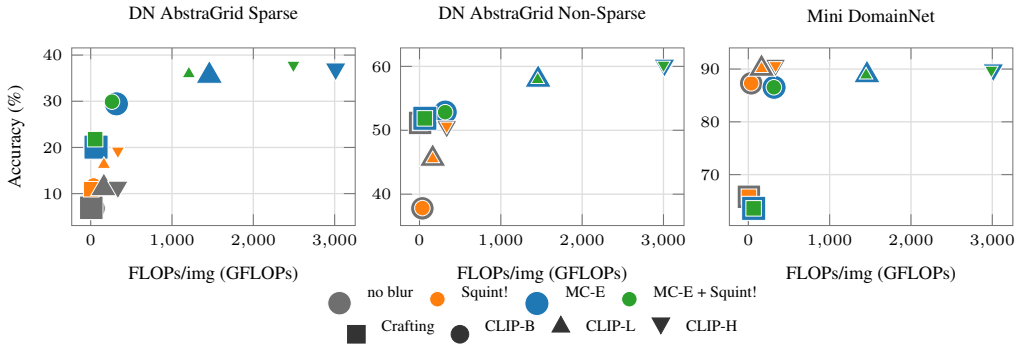

Figure 10: **Accuracy–computation trade-off.** Squint! improves sparse-domain accuracy with negligible additional cost, while Squint!+MC-E reduces the computational cost of MC-E while maintaining or improving its accuracy.

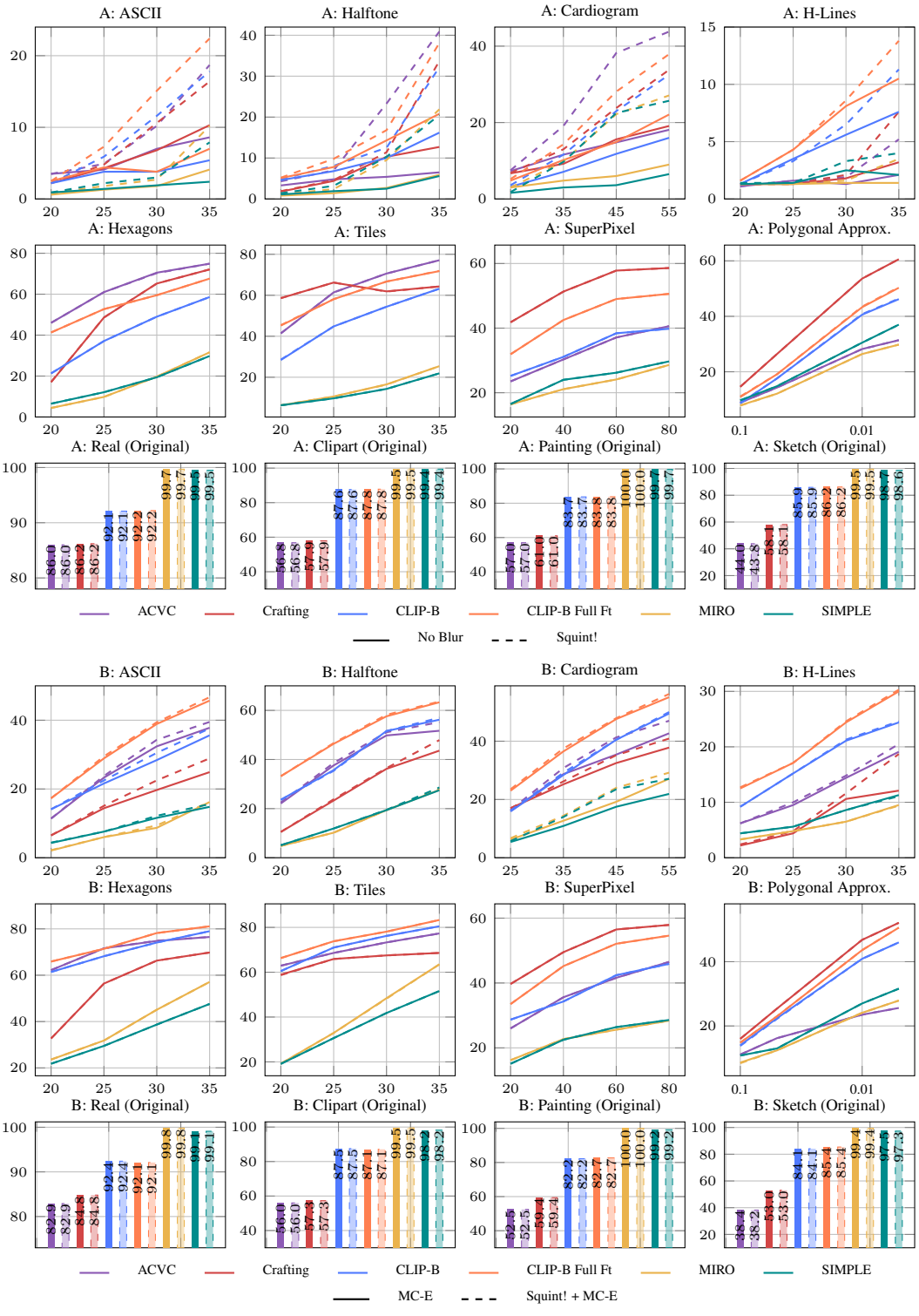

Figure 11: **Benchmarking domain generalization methods:** Sparse abstraction domains (ASCII, Halftone, Cardiogram, H-Lines) expose large performance drops relative to original domains, revealing a failure mode not captured by standard SSDG, MSDG, and zero-shot VLM evaluation. **A**: direct application of Squint! compared with no blur. **B**: comparison of the multi-blur baseline based on the prediction with the lowest entropy (MC-E), and Squint! + MC-E that uses Squint! for ensemble pruning.

	ResNet-18				L2D				ACVC				Crafting				SelfReg				MIRO				SIMPLE				ERM++			
	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!
Real	86.5	86.5	82.4	82.4	84.8	84.8	81.8	81.8	86.0	86.0	82.9	82.9	86.2	86.2	84.8	84.8	86.2	86.2	79.2	79.2	99.7	99.7	99.8	99.8	99.5	99.5	99.1	99.1	99.5	99.5	99.4	99.4
Clipart	47.0	47.0	45.7	45.7	53.0	53.0	52.5	52.5	56.8	56.8	56.0	56.0	57.9	57.9	57.3	57.3	52.4	52.4	48.7	48.7	99.5	99.5	99.5	99.5	99.4	99.4	98.2	98.2	99.7	99.7	99.1	99.1
Painting	51.6	51.6	49.0	49.0	55.6	55.6	54.1	54.1	57.0	57.0	52.5	52.5	61.0	61.0	59.4	59.4	53.8	53.8	47.3	47.3	100.0	100.0	100.0	100.0	99.7	99.7	99.2	99.2	100.0	100.0	99.4	99.4
Sketch	34.9	34.9	32.5	32.5	44.6	44.6	41.6	41.6	44.0	43.8	38.4	38.2	58.1	58.1	53.0	53.0	45.9	45.9	41.8	41.8	99.5	99.5	99.4	99.4	98.7	98.6	97.5	97.3	99.1	99.1	98.2	98.2
AS 20	1.4	1.4	1.6	1.6	1.4	1.6	1.4	1.4	3.5	3.5	11.4	11.4	2.2	2.2	6.5	6.5	1.4	1.4	1.1	1.1	0.6	0.8	2.1	2.2	0.9	0.9	4.3	4.4	0.6	0.8	1.6	1.6
AS 25	1.1	0.9	3.2	3.0	1.9	2.1	3.5	3.3	4.1	4.9	23.3	23.8	4.3	4.8	14.4	15.1	2.4	2.4	1.9	1.9	1.3	1.8	6.0	6.0	1.4	2.2	7.6	7.6	0.9	1.3	5.2	5.2
AS 30	2.4	2.5	5.2	5.9	2.5	3.2	4.9	5.6	7.0	10.2	32.4	34.3	6.8	10.6	19.7	22.5	3.8	3.8	4.6	4.8	1.8	2.7	8.7	9.4	1.9	3.0	11.6	12.2	1.4	2.4	7.9	7.9
AS 35	3.6	5.6	7.8	8.6	3.3	4.6	7.6	7.8	8.6	18.7	37.8	39.5	10.3	16.4	24.9	28.9	3.8	4.8	6.0	5.7	4.1	10.2	16.2	16.2	2.4	7.9	14.8	15.6	2.4	6.5	10.9	11.3
Ha 20	0.9	0.9	0.9	0.9	1.1	1.3	1.6	1.6	3.3	4.3	22.1	22.5	1.8	1.9	10.5	10.5	0.9	0.9	1.1	1.1	0.8	1.1	4.8	4.8	1.1	1.4	5.1	5.1	0.6	0.8	3.6	3.6
Ha 25	1.3	1.8	3.2	3.8	2.4	3.6	4.4	5.6	4.8	7.0	37.6	38.6	4.4	4.6	23.3	23.8	1.8	1.9	3.3	3.8	1.4	2.4	10.2	10.2	1.9	3.2	11.9	11.9	0.9	1.9	7.6	7.9
Ha 30	2.5	2.7	6.5	6.5	3.5	7.1	7.9	8.6	5.4	23.3	49.8	51.1	10.3	11.4	36.2	36.5	3.0	6.0	6.3	7.6	2.7	9.5	19.4	19.4	2.5	10.3	19.4	19.5	2.9	6.0	11.8	11.8
Ha 35	4.3	10.8	12.5	15.2	3.0	10.0	11.9	14.4	6.5	40.9	51.7	55.2	12.7	33.7	43.6	47.9	3.0	10.3	9.7	11.6	5.9	21.9	27.6	28.1	5.6	20.5	27.5	28.6	4.1	13.5	18.7	19.4
Ca 25	3.5	3.6	4.3	4.4	3.5	3.5	4.3	4.3	7.3	7.5	16.0	16.4	6.7	6.8	17.0	17.0	3.3	3.3	3.8	4.0	3.0	3.0	6.2	6.7	1.6	1.8	5.4	5.7	1.4	2.1	2.9	3.3
Ca 35	3.8	5.1	5.9	7.0	4.8	6.3	7.1	7.8	11.6	19.1	28.7	30.8	9.2	13.0	25.2	26.2	4.8	7.0	8.4	9.5	4.8	9.8	12.7	14.4	3.0	9.7	10.9	14.0	2.2	5.1	5.9	7.8
Ca 45	5.6	10.0	10.0	12.5	6.3	12.5	10.9	13.2	14.8	38.2	35.4	41.3	15.6	23.8	32.5	35.2	6.0	11.1	11.9	12.1	6.0	22.2	19.2	24.0	3.6	22.5	17.5	23.5	4.8	12.7	10.3	13.7
Ca 55	6.7	14.6	14.8	17.3	7.5	16.4	16.2	16.8	18.1	43.8	42.7	47.0	19.1	33.8	37.8	40.9	8.7	15.1	16.8	15.9	9.0	27.1	27.1	29.2	6.5	25.7	21.9	27.1	6.3	17.1	14.9	18.7
HL 20	0.9	0.9	1.8	1.8	1.1	1.1	1.6	1.6	1.1	1.1	6.2	6.2	1.3	1.3	2.2	2.4	1.3	1.3	1.3	1.3	1.3	1.3	3.3	3.3	1.3	1.4	4.4	4.4	1.6	1.6	2.7	2.9
HL 25	1.8	1.8	2.4	2.4	1.4	1.4	2.9	2.9	1.6	1.6	9.5	10.0	1.3	1.3	4.4	4.8	1.6	1.4	1.3	1.3	1.3	1.3	4.8	4.8	1.4	1.4	5.6	5.6	1.3	1.3	3.6	3.8
HL 30	0.8	0.8	3.8	3.8	1.6	1.6	4.0	4.3	1.3	1.9	14.4	14.8	1.8	2.1	10.6	11.6	1.3	1.3	2.2	2.2	1.4	1.6	6.5	6.5	2.5	3.3	8.6	8.6	0.9	0.9	5.6	5.6
HL 35	1.8	2.4	4.9	5.4	2.2	2.9	6.8	8.6	2.1	5.2	19.1	20.5	3.2	7.6	12.1	18.7	2.2	2.5	5.1	6.0	1.4	3.5	9.5	9.4	2.1	4.0	11.3	11.1	0.9	1.8	8.7	8.9
He 20	1.1	1.1	6.5	6.5	9.4	9.4	15.7	15.7	46.0	46.0	62.2	62.2	17.0	17.1	32.7	32.7	3.3	3.3	7.0	7.0	4.4	4.4	23.6	23.6	6.5	6.5	21.8	21.8	3.5	3.5	13.5	13.5
He 25	5.2	5.2	17.6	17.6	16.0	16.0	26.7	26.7	61.0	61.0	71.6	71.6	48.6	48.6	56.4	56.4	8.2	8.2	13.2	13.2	9.8	9.8	31.8	31.8	12.1	12.1	29.5	29.5	7.0	7.0	20.3	20.3
He 30	10.6	10.6	27.6	27.6	25.1	25.1	37.5	37.5	70.5	70.3	74.8	74.9	65.2	65.2	66.3	66.3	17.0	17.0	21.9	21.9	19.7	19.7	45.1	45.1	19.4	19.4	38.7	38.7	16.4	16.4	30.5	30.5
He 35	19.8	19.8	39.2	39.2	33.3	33.3	45.1	45.1	74.9	74.9	76.5	76.5	72.1	72.1	69.8	69.8	24.0	23.8	31.3	31.3	31.6	31.6	57.1	57.1	29.8	29.8	47.6	47.6	28.1	28.1	41.1	41.1
Ti 20	6.8	6.8	19.8	19.8	5.7	5.7	14.3	14.3	41.4	41.4	62.9	62.9	58.6	58.6	58.9	58.7	4.9	4.9	7.6	7.6	6.3	6.3	19.2	19.2	6.3	6.3	19.1	19.1	2.5	2.5	13.3	13.3
Ti 25	10.8	10.8	30.0	30.0	11.4	11.4	24.3	24.3	61.4	61.4	68.6	68.6	66.2	66.2	65.9	65.9	8.4	8.4	15.4	15.4	10.6	10.6	32.9	32.9	9.7	9.7	30.6	30.6	4.4	4.4	22.4	22.4
Ti 30	14.3	14.3	40.8	40.8	17.9	17.9	35.6	35.6	70.6	70.6	73.3	73.3	61.9	61.8	67.5	67.5	13.2	13.2	23.8	23.8	16.5	16.4	48.4	48.2	14.3	14.3	41.8	41.8	8.9	8.7	34.0	34.0
Ti 35	20.9	20.9	49.7	49.7	23.5	23.5	45.6	45.6	77.1	77.1	77.3	77.3	64.3	64.3	68.6	68.6	19.5	19.5	36.4	36.4	25.4	25.4	63.5	63.5	21.9	21.9	51.6	51.6	14.6	14.6	46.2	46.2
SP 20	13.5	13.5	14.6	14.6	22.9	22.9	19.7	19.7	23.5	23.5	26.0	26.0	41.8	41.8	39.7	39.7	15.2	15.2	14.1	14.1	16.4	16.4	16.2	16.2	16.5	16.5	15.1	15.1	17.0	17.0	16.0	16.0
SP 40	19.1	19.1	21.1	21.1	32.1	32.1	30.0	30.0	30.3	30.3	35.6	35.6	51.3	51.3	49.5	49.5	20.9	20.9	20.2	20.2	21.1	21.1	22.7	22.7	24.0	24.0	22.5	22.5	20.9	20.9	21.9	21.9
SP 60	20.6	20.6	24.4	24.4	36.4	36.4	34.9	34.9	37.1	37.1	41.6	41.6	57.8	57.8	56.5	56.5	22.5	22.5	21.6	21.6	24.1	24.1	25.6	25.6	26.2	26.2	26.4	26.4	25.4	25.4	25.6	25.6
SP 80	24.0	24.0	27.5	27.5	39.5	39.5	37.1	37.1	40.6	40.6	46.5	46.5	58.6	58.6	57.9	57.9	24.4	24.4	23.6	23.6	28.6	28.6	28.4	28.4	29.7	29.7	28.6	28.6	27.3	27.3	26.7	26.7
PA 0.1	4.4	4.4	6.2	6.2	10.5	10.5	10.0	10.0	8.9	8.9	11.1	11.1	14.6	14.6	16.0	16.0	10.0	10.0	8.4	8.4	7.9	7.8	8.6	8.4	9.7	9.7	10.8	10.8	7.8	7.6	9.2	9.0
PA 0.05	7.1	7.1	8.4	8.4	15.2	15.2	16.0	16.0	14.3	14.3	16.2	16.2	26.4	26.4	25.4	25.4	14.3	14.3	12.2	12.2	12.1	12.1	12.4	12.4	14.8	14.8	13.0	13.0	12.5	12.5	12.2	12.2
PA 0.01	14.9	14.9	14.6	14.6	28.6	28.6	27.5	27.5	28.2	28.2	23.5	23.5	53.6	53.6	46.8	46.8	25.7	25.7	21.3	21.3	26.4	26.4	24.1	24.1	30.5	30.5	27.0	27.0	29.0	29.0	25.7	25.7
PA 0.005	16.4	16.4	16.0	16.0	30.8	30.8	28.7	28.7	31.4	31.4	25.6	25.6	60.6	60.6	52.1	52.1	29.2	29.2	24.3	24.3	29.8	29.8	27.9	27.9	37.0	37.0	31.6	31.6	35.9	35.9	29.2	29.2

Table 5: **DG Tables.**

	CLIP ViT-B-16				+ fine-tuned blur				+ fine-tuned imgaug				+ fine-tuned full				CLIP ViT-L-14				CLIP ViT-H-14				SigLIP ViT-B-16				SigLIP ViT-L-16-384			
	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!
Real	92.1	92.1	92.4	92.4	91.9	91.9	91.4	91.4	92.2	92.2	92.7	92.7	92.2	92.2	92.1	92.1	95.1	95.1	93.8	93.8	95.2	95.2	93.5	93.5	94.6	94.6	95.1	95.1	95.9	95.9	95.9	95.9
Clipart	87.6	87.6	87.5	87.5	87.5	87.5	87.0	87.0	87.8	87.8	86.8	86.8	87.8	87.8	87.1	87.1	90.2	90.2	89.7	89.7	90.8	90.8	90.6	90.6	92.2	92.2	92.4	92.4	93.2	93.2	93.5	93.5
Painting	83.7	83.7	82.2	82.2	83.7	83.7	82.7	82.7	84.1	84.1	82.7	82.7	83.8	83.8	82.7	82.7	86.0	86.0	84.6	84.6	87.3	87.1	86.4	86.4	87.8	87.8	84.8	84.8	91.6	91.6	90.3	90.3
Sketch	85.9	85.9	84.1	84.1	86.4	86.4	84.4	84.4	86.2	86.2	85.4	85.4	86.2	86.2	85.4	85.4	89.0	89.0	87.3	87.3	89.7	89.5	88.9	88.7	89.8	89.8	90.3	90.3	92.1	92.1	92.5	92.5
AS 20	2.2	2.2	14.1	14.1	2.2	2.2	14.1	14.1	2.7	2.7	17.9	17.9	2.5	2.5	17.3	17.3	3.2	3.2	16.0	16.0	3.2	5.9	20.3	20.6	4.0	4.8	17.1	17.3	8.1	8.1	20.8	20.8
AS 25	3.8	5.9	21.6	22.5	3.8	6.2	23.0	23.5	4.3	6.8	28.7	29.4	4.4	7.3	28.9	29.5	4.1	7.9	25.2	25.7	4.6	9.7	29.2	30.0	4.9	7.9	24.0	24.4	9.8	10.3	31.8	31.9
AS 30	3.8	11.6	28.4	30.5	4.0	11.8	31.8	34.0	3.6	15.4	37.9	39.1	3.8	15.1	38.9	39.4	5.4	16.5	34.6	35.9	6.7	22.4	37.8	39.2	6.5	15.2	31.6	32.4	14.8	14.9	42.1	42.1
AS 35	5.4	17.8	35.6	37.6	5.2	18.4	39.5	41.3	7.1	22.2	45.2	46.7	7.0	22.4	45.7	46.7	8.7	23.2	44.1	45.2	9.5	27.3	46.7	48.4	10.9	22.2	38.9	39.2	18.4	21.9	52.5	52.5
Ha 20	4.6	4.8	23.6	23.6	4.3	4.4	27.0	27.0	5.1	5.2	33.2	33.2	5.1	5.2	33.2	33.2	6.3	6.5	32.1	32.1	8.9	14.8	31.6	33.0	7.8	8.4	21.9	21.9	14.9	14.9	34.9	34.9
Ha 25	6.8	8.1	35.7	35.4	6.7	8.2	39.4	39.2	7.8	9.8	46.2	46.2	7.8	9.8	46.4	46.7	13.7	15.1	43.8	43.8	13.7	16.4	44.0	44.4	13.3	17.8	35.9	35.4	23.2	24.3	53.3	53.5
Ha 30	10.0	12.5	51.6	51.6	10.2	12.7	52.7	52.9	14.1	16.8	57.6	58.1	14.3	16.8	57.6	58.2	19.1	20.2	54.1	54.3	20.2	25.6	56.4	57.3	17.3	29.5	49.7	50.9	31.3	31.9	64.0	63.8
Ha 35	16.2	32.2	56.2	57.0	16.4	32.5	57.8	57.9	21.3	37.6	63.5	63.3	20.8	38.1	63.3	63.6	26.5	44.4	62.4	64.1	28.1	36.2	64.8	65.2	23.5	44.8	55.7	56.2	36.2	41.6	70.5	70.3
Ca 25	3.2	3.6	15.9	16.2	3.8	4.3	18.7	19.2	4.8	5.2	20.8	21.1	4.8	5.2	23.0	23.5	4.6	6.2	24.3	24.9	4.6	7.5	25.4	26.5	7.3	8.6	23.0	23.3	8.6	8.6	32.5	32.5
Ca 35	7.1	11.6	28.4	29.2	6.8	11.3	31.4	31.9	9.7	14.0	33.0	34.1	10.2	14.3	36.5	37.5	12.9	16.5	40.8	41.1	10.5	17.6	39.8	40.5	14.0	23.2	39.4	40.5	16.8	17.0	49.8	49.8
Ca 45	11.8	23.0	40.5	40.5	11.6	22.7	42.9	42.5	14.4	26.2	45.6	46.4	14.9	28.1	47.6	47.9	20.6	29.5	52.2	52.5	18.9	35.1	50.6	50.9	23.2	46.0	50.8	50.2	24.3	26.7	60.0	60.2
Ca 55	16.0	32.5	49.5	50.0	16.7	33.3	50.5	50.6	21.6	37.5	54.1	54.1	22.1	37.9	55.1	56.2	30.6	40.9	57.9	57.8	27.5	46.4	57.3	58.2	30.8	51.9	57.8	56.5	35.4	37.1	66.5	66.7
HL 20	1.3	1.3	9.2	9.2	1.1	1.1	10.6	10.6	1.6	1.8	12.5	12.5	1.6	1.6	12.5	12.7	1.4	1.8	10.0	10.0	1.6	5.1	11.3	11.8	3.3	3.6	9.8	10.0	3.5	3.8	9.8	10.0
HL 25	3.5	3.3	15.2	15.2	3.5	3.6	14.6	14.6	4.1	4.0	17.1	17.1	4.3	4.3	17.1	17.1	4.3	4.9	17.1	17.3	4.8	7.3	19.4	20.2	4.6	4.9	16.7	16.8	7.6	8.2	24.3	24.4
HL 30	5.6	6.5	21.1	21.3	6.0	7.0	22.7	22.9	8.4	8.7	24.1	24.3	8.1	8.6	24.4	24.6	7.3	7.8	23.8	23.8	7.5	10.8	25.2	26.2	8.1	8.6	26.7	26.7	8.4	14.3	30.3	31.4
HL 35	7.6	11.3	24.4	24.6	7.9	11.6	25.1	25.6	10.8	14.8	27.3	27.8	10.5	13.8	30.0	30.3	10.5	16.2	30.0	30.5	12.7	20.0	32.1	32.7	12.2	19.8	30.2	30.9	12.7	13.8	33.5	33.3
He 20	21.3	21.3	61.4	61.4	22.2	22.2	63.5	63.5	38.9	38.9	65.4	65.4	41.3	41.3	65.9	65.9	32.2	32.2	68.4	68.4	45.6	45.6	70.6	70.8	41.4	41.4	61.4	61.4	54.1	54.1	73.2	73.2
He 25	37.1	37.1	68.2	68.2	39.1	39.1	69.0	69.0	51.4	51.4	72.5	72.5	52.7	52.7	71.4	71.4	48.6	48.6	78.4	78.4	58.6	58.6	78.9	78.9	56.0	56.0	71.8	71.8	68.7	68.7	80.5	80.5
He 30	49.0	49.0	74.1	74.1	49.0	49.0	75.6	75.6	58.6	58.6	77.5	77.5	59.5	59.5	78.2	78.2	60.0	60.0	82.7	82.7	67.8	67.9	83.3	83.3	68.1	68.1	77.1	77.1	72.4	72.4	85.7	85.7
He 35	58.6	58.6	79.0	79.0	58.9	58.9	80.0	80.0	67.0	67.0	80.6	80.6	67.6	67.6	81.1	81.1	68.4	68.4	85.9	85.9	74.6	74.8	87.3	87.3	71.0	71.0	81.9	81.9	80.2	80.2	88.4	88.4
Ti 20	28.4	28.6	60.5	60.5	28.6	28.7	65.7	65.7	45.4	45.6	67.0	67.0	45.2	45.4	66.3	66.3	39.4	39.5	67.3	67.3	48.9	48.9	69.4	69.4	48.4	48.4	61.6	61.6	59.1	59.1	75.1	75.1
Ti 25	44.8	44.8	71.0	71.0	45.6	45.6	72.7	72.7	58.7	58.7	74.9	74.9	58.1	58.1	73.8	73.8	53.3	53.3	76.0	76.0	58.7	59.1	80.0	80.0	60.8	60.8	71.1	71.1	72.5	72.5	81.8	81.8
Ti 30	54.4	54.4	76.2	76.2	54.6	54.6	77.3	77.3	66.7	66.7	78.9	78.9	66.7	66.7	78.1	78.1	64.8	64.8	82.4	82.4	70.0	70.2	84.8	84.8	69.5	69.5	78.1	78.1	77.9	77.9	86.2	86.2
Ti 35	63.2	63.2	80.5	80.5	63.2	63.3	82.5	82.5	72.2	72.4	82.9	82.9	71.8	71.9	83.3	83.3	72.9	72.9	87.3	87.3	77.3	77.3	87.9	87.9	75.4	75.4	81.3	81.3	85.1	85.1	89.4	89.4
SP 20	25.2	25.2	28.7	28.7	24.9	24.9	28.4	28.4	31.1	31.1	32.2	32.2	31.9	31.9	33.5	33.5	28.7	28.7	28.7	28.7	28.2	28.2	31.3	31.3	29.4	29.4	31.3	31.3	35.4	35.4	36.0	36.0
SP 40	31.1	31.1	34.3	34.3	31.4	31.4	34.4	34.4	41.4	41.4	43.8	43.8	42.5	42.5	45.2	45.2	34.6	34.6	39.2	39.2	40.0	40.0	41.6	41.6	36.0	36.0	38.9	38.9	43.5	43.5	46.0	46.0
SP 60	38.4	38.4	42.4	42.4	38.1	38.1	43.0	43.0	47.1	47.1	51.6	51.6	49.0	49.0	52.1	52.1	44.4	44.4	46.7	46.7	48.4	48.4	48.4	48.4	43.3	43.3	47.5	47.5	50.8	50.8	54.0	54.0
SP 80	39.8	39.8	45.9	45.9	39.5	39.5	46.8	46.8	49.7	49.7	53.3	53.3	50.6	50.6	54.6	54.6	47.3	47.3	48.9	48.9	48.9	48.9	51.4	51.4	47.8	47.8	50.9	50.9	56.4	56.4	57.0	57.0
PA 0.1	8.6	8.7	14.0	13.8	10.2	10.3	14.4	14.4	10.8	10.9	14.8	14.8	10.9	11.1	14.8	14.8	13.2	13.0	14.4	14.6	13.7	14.0	16.5	16.5	14.4	14.4	17.3	17.1	16.7	16.7	19.4	19.4
PA 0.05	17.6	17.6	22.4	22.2	18.6	18.6	22.9	22.7	18.4	18.4	23.8	23.8	19.1	19.1	23.0	23.0	20.6	20.5	24.8	24.8	23.0	23.3	26.8	26.8	26.5	26.5	29.0	29.0	28.1	28.1	30.9	30.9
PA 0.01	40.6	40.8	40.9	40.9	40.8	40.9	41.6	41.6	43.3	43.5	44.0	44.0	43.3	43.5	43.6	43.6	47.3	47.1	45.6	45.4	49.0	49.0	48.7	48.7	51.6	51.6	50.5	50.5	57.6	57.6	61.6	61.6
PA 0.005	46.2	46.4	46.0	46.0	48.1	48.2	46.5	46.5	50.3	50.5	50.2	50.2	50.3	50.5	50.6	50.6	53.3	53.2	50.2	50.0	56.4	56.4	56.0	56.0	56.8	56.8	54.6	54.6	64.8	64.8	65.1	65.1

Table 6: **VLM Tables 1/2.**

	SigLIP ViT-SO400M-14-384				SigLIP2 ViT-B-16-256				SigLIP2 ViT-L-16-256				SigLIP2 ViT-SO400M-16-384				PE-Core B-16				PE-Core L-14-336				PE-Core bigG-14-448				Supervised			
	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!	no blur	Squint!	MC-E	MC-E + Squint!
Real	96.0	96.0	95.7	95.7	94.8	94.8	94.3	94.3	96.0	96.0	94.9	94.9	95.9	95.9	96.0	96.0	94.1	94.1	93.0	93.0	95.6	95.6	95.4	95.4	95.9	95.9	95.9	95.9	92.1	92.1	92.4	92.4
Clipart	94.0	94.0	93.8	93.8	94.0	94.0	93.5	93.5	93.3	93.3	93.2	93.2	94.0	94.0	93.3	93.3	90.8	90.8	90.8	90.8	91.8	91.8	92.9	92.9	93.5	93.5	93.5	93.5	87.6	87.6	87.5	87.5
Painting	91.4	91.4	91.6	91.6	89.7	89.7	88.2	88.2	91.0	91.0	90.2	90.2	92.2	92.2	91.6	91.6	85.9	85.9	84.9	84.9	90.3	90.3	90.8	90.8	92.5	92.5	92.2	92.2	83.7	83.7	82.2	82.2
Sketch	93.3	93.2	92.9	92.9	89.8	89.8	90.2	90.2	92.4	92.4	93.0	93.0	92.7	92.7	92.5	92.5	89.2	89.0	87.3	87.3	91.8	91.8	91.1	91.1	93.3	93.2	93.3	93.2	85.9	85.9	84.1	84.1
AS 20	9.0	9.0	17.9	17.9	4.6	4.9	17.9	17.9	7.8	8.2	25.7	25.9	7.3	7.3	23.8	23.8	4.3	9.0	26.2	26.4	8.6	8.7	36.7	36.7	20.0	20.2	49.4	49.4	2.2	2.4	14.1	14.0
AS 25	12.1	12.9	29.4	29.5	7.5	11.9	24.6	26.0	9.8	15.9	41.1	42.1	11.3	12.4	41.4	41.3	7.9	14.6	39.1	38.9	15.1	17.5	51.7	51.9	31.1	31.1	65.1	65.2	3.8	4.9	21.6	22.4
AS 30	15.1	18.6	42.4	43.0	8.4	13.5	36.7	37.9	15.7	21.4	53.5	54.6	14.3	20.2	53.2	53.2	10.6	40.0	51.6	53.2	19.8	28.7	64.6	64.9	42.5	43.6	73.0	73.0	3.8	6.8	28.4	30.0
AS 35	18.9	27.5	52.5	53.2	12.7	24.0	43.0	43.8	19.5	38.1	62.5	63.5	22.9	32.2	63.6	64.0	14.1	44.1	59.8	60.2	29.2	40.8	70.5	71.3	50.0	53.2	79.0	79.2	5.4	10.2	35.6	37.1
Ha 20	18.1	17.9	32.1	32.2	13.3	13.7	30.6	30.8	20.3	20.9	53.6	53.6	16.7	16.8	45.9	45.9	13.5	19.4	44.0	44.9	33.2	35.1	61.6	61.6	62.2	62.4	74.8	74.8	4.6	4.6	23.6	23.6
Ha 25	23.8	25.4	51.6	51.4	18.6	19.2	39.8	40.2	30.6	34.6	61.1	60.5	28.1	29.7	65.9	66.0	21.9	30.8	59.4	59.7	48.9	51.4	74.8	74.8	71.6	71.3	81.8	81.8	6.8	7.9	35.7	35.9
Ha 30	31.6	35.2	65.6	65.4	23.6	26.7	54.8	55.1	38.7	42.2	70.8	71.0	37.3	41.9	74.1	74.3	33.0	50.5	70.8	71.3	60.0	66.2	80.5	80.8	76.7	77.1	87.3	87.8	10.0	11.1	51.6	51.1
Ha 35	39.1	46.8	71.9	72.5	28.6	43.0	63.8	66.0	47.5	60.8	77.8	77.9	46.5	58.1	78.1	78.2	38.2	59.1	74.1	73.3	67.6	70.3	84.9	85.2	80.8	83.3	90.6	90.8	16.2	39.7	56.2	56.4
Ca 25	9.7	12.4	30.2	31.6	8.1	9.5	28.2	28.6	14.4	16.2	41.8	42.5	10.2	15.1	42.1	42.2	6.8	10.5	37.9	38.1	20.8	25.2	51.3	52.5	32.1	32.5	57.1	57.1	3.2	3.8	15.9	15.9
Ca 35	16.7	25.7	51.9	52.7	14.1	19.1	47.9	48.1	25.1	28.9	60.8	60.8	23.6	31.6	61.9	62.1	14.1	28.1	52.7	53.8	36.4	37.8	64.0	64.1	47.6	50.3	73.7	74.3	7.1	7.6	28.4	28.4
Ca 45	27.9	32.5	62.7	62.7	23.5	35.6	58.6	58.1	36.7	45.4	69.5	68.2	32.2	35.7	70.6	70.2	23.2	43.6	60.8	61.1	49.7	61.1	73.0	73.5	57.9	58.7	79.5	79.5	11.8	11.8	40.5	40.3
Ca 55	37.0	47.6	68.7	69.5	31.1	58.6	65.4	66.2	45.7	65.2	73.5	73.3	40.8	51.1	75.4	75.2	31.1	54.3	65.6	65.4	56.4	65.2	77.5	78.1	67.6	68.2	82.9	82.5	16.0	16.2	49.5	49.7
HL 20	2.9	3.2	9.0	9.0	3.0	6.2	11.6	12.1	4.1	10.2	14.0	16.8	3.6	3.6	14.1	14.1	1.6	5.9	17.5	18.6	7.0	7.8	20.8	20.9	8.7	10.8	36.0	36.0	1.3	1.4	9.2	9.2
HL 25	4.4	6.3	17.0	17.0	5.6	7.0	17.1	17.5	8.6	11.1	26.2	26.4	11.9	13.5	29.4	29.4	4.4	9.4	27.3	28.2	18.1	22.1	38.7	38.9	19.2	18.9	48.2	47.9	3.5	7.0	15.2	15.9
HL 30	9.7	12.9	29.7	30.0	7.1	9.0	24.8	24.9	10.6	12.5	32.2	32.2	14.3	19.4	37.9	38.2	8.6	13.2	35.2	35.1	23.0	28.7	46.0	47.6	32.5	34.8	61.3	61.4	5.6	6.7	21.1	20.8
HL 35	12.4	19.1	37.8	38.9	8.4	13.7	31.8	32.1	13.3	17.6	37.6	38.2	20.2	26.5	43.6	44.3	14.0	28.1	41.6	42.2	29.0	29.8	53.0	53.0	41.1	45.4	66.3	66.3	7.6	8.6	24.4	24.6
He 20	54.6	54.6	74.8	74.8	46.7	46.7	67.9	67.9	58.6	58.6	77.1	77.1	57.5	57.5	77.9	77.9	55.6	55.6	74.9	74.9	74.1	74.1	83.5	83.5	83.8	83.8	88.9	88.9	21.3	21.3	61.4	61.4
He 25	68.2	68.2	83.3	83.3	60.6	60.6	77.1	77.1	71.4	71.4	84.1	84.1	72.2	72.2	83.7	83.7	64.9	64.9	81.1	81.1	81.6	81.6	87.5	87.5	88.4	88.4	91.8	91.8	37.1	37.3	68.2	68.2
He 30	75.1	75.1	86.5	86.4	69.0	69.0	82.1	82.1	77.1	77.1	87.1	87.1	78.1	78.1	87.3	87.3	75.2	75.2	84.9	84.9	87.0	87.0	90.6	90.6	92.1	92.1	93.8	93.8	49.0	49.0	74.1	74.1
He 35	80.8	80.8	90.2	90.2	75.9	76.0	86.4	86.4	82.7	82.5	90.6	90.6	84.0	84.0	90.6	90.6	77.9	77.9	85.9	85.9	89.0	89.0	91.8	91.8	93.7	93.7	95.1	95.1	58.6	58.6	79.0	79.0
Ti 20	65.4	65.2	74.1	74.1	55.1	55.1	66.2	66.2	64.0	64.0	73.8	73.8	66.8	67.0	75.6	75.6	74.8	74.8	78.4	78.4	84.0	84.0	86.5	86.5	89.5	89.5	90.0	90.0	28.4	28.6	60.5	60.3
Ti 25	73.5	73.5	81.8	81.8	65.6	65.6	75.9	75.9	73.3	73.3	81.9	81.9	77.9	77.9	83.2	83.2	80.8	80.8	82.4	82.5	87.9	87.9	89.2	89.2	93.2	93.2	93.3	93.3	44.8	44.8	71.0	70.8
Ti 30	79.8	80.0	87.1	87.1	72.1	72.1	81.9	81.9	80.3	80.3	85.7	85.7	82.7	82.7	87.0	87.0	85.1	85.2	85.2	85.2	91.6	91.6	90.8	90.8	94.3	94.3	94.1	94.1	54.4	54.8	76.2	76.2
Ti 35	86.8	86.8	90.5	90.5	79.4	79.4	85.4	85.4	86.7	86.7	89.0	89.0	87.3	87.3	90.0	90.0	87.3	87.3	87.0	87.0	93.0	93.0	92.2	92.2	94.9	94.9	95.2	95.2	63.2	63.0	80.5	80.5
SP 20	37.5	37.3	37.1	37.0	32.9	32.9	34.6	34.6	35.2	35.2	35.6	35.6	38.1	38.1	37.0	37.0	33.7	33.7	33.8	33.8	46.0	46.0	44.1	44.1	50.3	50.3	50.2	50.2	25.2	25.2	28.7	28.7
SP 40	45.7	45.7	47.3	47.3	40.5	40.5	43.0	43.0	46.2	46.2	46.7	46.7	49.7	49.7	50.8	50.6	44.8	44.8	44.8	44.8	55.2	55.1	54.0	53.8	64.6	64.6	63.3	63.3	31.1	31.1	34.3	34.3
SP 60	53.5	53.5	55.2	55.2	48.2	48.2	50.8	50.8	53.3	53.3	53.6	53.6	56.2	56.2	57.5	57.5	50.9	50.9	52.9	52.9	63.3	63.3	62.5	62.5	69.7	69.7	69.2	69.2	38.4	38.4	42.4	42.4
SP 80	56.8	56.8	57.6	57.6	49.7	49.7	53.3	53.3	54.4	54.4	57.5	57.5	60.2	60.2	62.1	62.1	53.0	53.0	56.0	56.0	66.8	66.8	66.3	66.3	72.5	72.5	72.2	72.2	39.8	39.8	45.9	45.9
PA 0.1	15.4	15.4	17.8	17.9	16.0	15.9	18.4	18.4	15.4	15.4	19.1	19.1	17.0	17.0	20.8	20.8	13.3	13.0	15.9	16.0	17.0	16.8	20.9	20.9	24.1	24.1	27.6	27.6	8.6	8.6	14.0	14.0
PA 0.05	29.7	29.7	32.2	32.2	26.5	26.4	27.9	27.9	27.5	27.5	29.8	29.8	27.3	27.3	32.9	32.9	21.3	20.9	23.2	22.9	30.2	30.2	33.8	33.8	37.5	37.5	41.4	41.4	17.6	17.6	22.4	22.4
PA 0.01	60.5	60.5	63.0	63.0	54.0	53.8	54.9	54.9	59.4	59.4	60.0	60.0	61.0	60.8	63.2	63.2	50.0	50.2	50.8	50.6	63.5	63.5	62.9	62.9	68.2	68.4	72.2	72.2	40.6	40.8	40.9	40.9
PA 0.005	66.2	66.0	66.0	65.9	60.0	59.8	60.0	60.0	63.6	63.6	63.0	63.0	66.8	66.8	68.4	68.4	55.1	54.9	55.9	55.9	69.0	69.2	67.8	67.8	75.2	74.9	76.0	76.0	46.2	46.4	46.0	46.0

Table 7: **VLM Tables 2/2.**

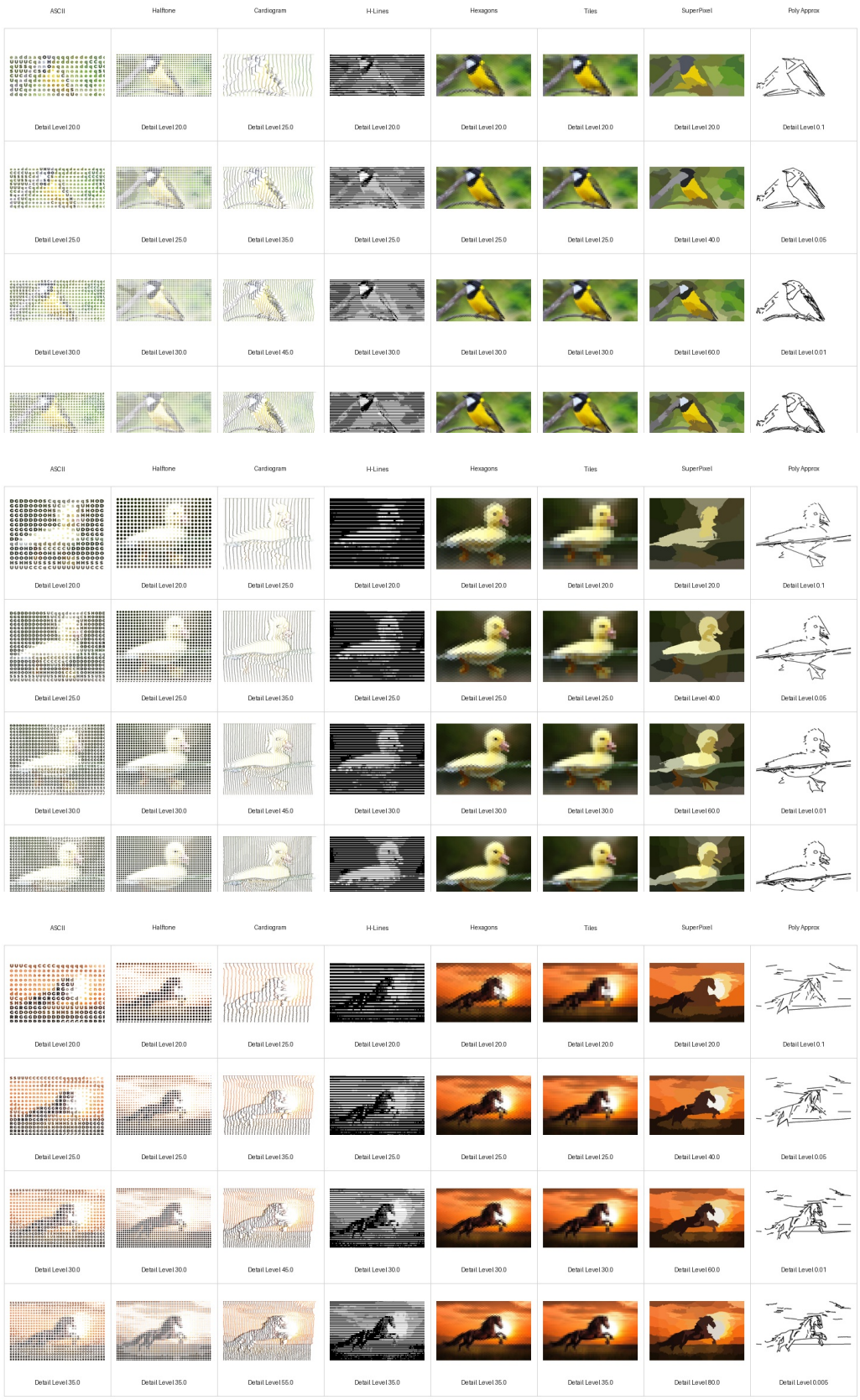

Figure 12: Examples of DN AbstraGrid. A bird, a duck, and a horse.

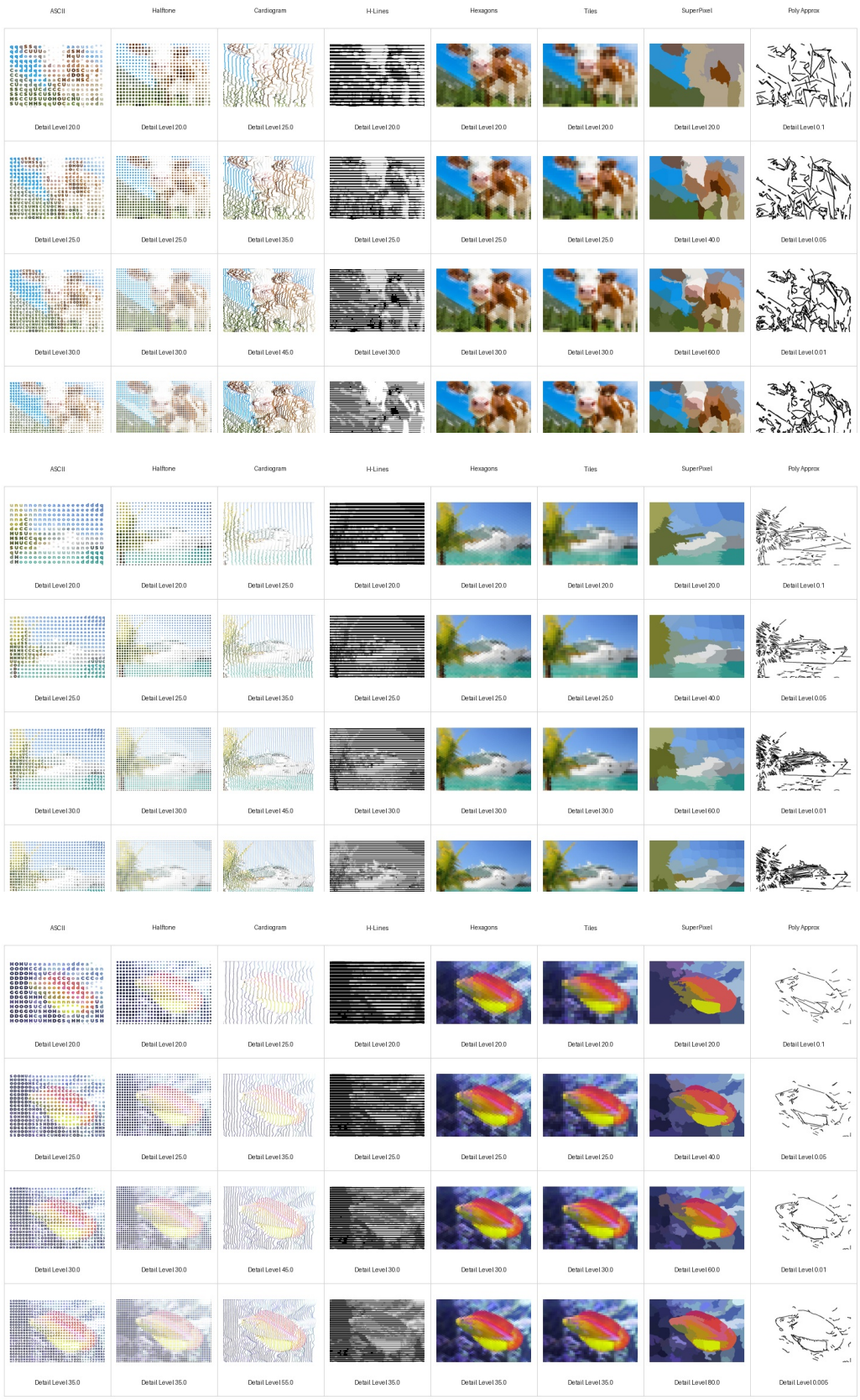

Figure 13: Examples of DN AbstraGrid. A cow, a cruise ship, and a fish.