Supplementary Material for Revisiting Vision–Language Foundations for No-Reference Image Quality Assessment

A. Extended Finetuning and Activation-Function Study

A.1. Finetuning Discussion

We fix the LoRA rank to 4 in all experiments based on our ablation study (Fig. S4) that indicates that rank-4 adapters converge noticeably faster than both lower and higher ranks and achieve this efficiency with minimal memory and compute overhead. For the ResNet encoder, we attach LoRA adapters to the convolutional layers inside each residual block. We choose these layers because they govern spatial filtering and channel mixing, giving high leverage per parameter.

Table S1 contrasts two regimes: (i) our full setup with rank-4 LoRA adapters injected into the query and key projections, and (ii) a variant that keeps the visual backbone frozen while training only the lightweight MLP regressor. LoRA fine-tuning yields consistent gains in SRCC/PLCC across all MLP designs and on six of the seven benchmarks. The single exception is FLIVE with the first activation of the MLP head swapped with sigmoid, where the frozen-backbone variant performs slightly better than the LoRA variant. We suspect that the sigmoid activation is the bottleneck that saturates on its 39 K-image scale, capping the head's capacity. Saturation suppresses gradient flow, so the LoRA adapters cannot harvest their usual gains. We also assess full fine-tuning with both activation heads (baseline and sigmoid). The sigmoid configuration yields slight improvements relative to LoRA, but given LoRA's markedly better efficiency, we standardize on LoRA for subsequent experiments.

Full Finetuning FT For the full finetuning experiments, we keep all other settings identical to the LoRA finetuning configuration but reduce the backbone learning rate to 5×10^{-6} and set the MLP head learning rate to 1×10^{-4}. This conservative schedule mitigates degradation of the pretrained encoder representations and makes the experiments comparable.

Embedding extraction. Unless otherwise stated, we follow the default Hugging Face (HF) implementations and use the encoder's pooled representation exposed by the model's forward pass. *CLIP/SigLIP:* we call the vision tower and take the projected image embedding (`image_embeds`), i.e., pooled visual features passed through the model's projection head. *DINOv2/DINOv3:* we average the last hidden-state tokens (global mean over patch tokens) to obtain a single image vector. *Perception Encoder (ViT-L/14-336):* we use the pooled output provided by the HF checkpoint, followed by its projection layer when available. *ResNet-152:* we take the `pooler_output`, i.e., the global-average-pooled convolutional features returned by `ResNetModel`. For Diffusion Backbone, We use CleanDIFT[6] checkpoints and adopt the DP-IQA[2] feature-adapter recipe to aggregate UNet diffusion features into a fixed-length image embedding. All embeddings are then fed to the same prediction head.

A.2. Activation Function Analysis

We systematically evaluated all pairwise combinations of four common nonlinearities, Sigmoid, Leaky ReLU, GELU, and Tanh, in the two hidden layers of our three-layer MLP (Table 6, Main Paper). The Sigmoid $\rightarrow$ Leaky ReLU sequence yields the highest average SRCC and PLCC across the seven NR-IQA benchmarks, while the Sigmoid $\rightarrow$ Sigmoid variant performs marginally better on a few datasets. Because stacked sigmoids are prone to vanishing gradients, especially under high-data regimes. Hence, we opt for the more stable Sigmoid $\rightarrow$ Leaky ReLU configuration.

GELU offers no statistically significant advantage over Leaky ReLU yet incurs a higher computational cost due to its Gaussian error function; we therefore retain Leaky ReLU as the default second-layer gate. Tanh lags behind all other activations, a trend that is visually corroborated by the fragmented class clusters in the t-SNE embedding of Figure S2.

We also tested the reverse ordering (Leaky ReLU $\rightarrow$ Sigmoid, results omitted for brevity); this arrangement negates the convergence benefits provided by the initial sigmoid and does not improve final SRCC Scores. Future work will extend this analysis to recently proposed activations such as Swish [5] and Mish [4] to further probe their effect on quality-aware feature learning.

A.3. Learning-Rate Scheduling Strategy for Medium-Scale Datasets

For all medium-scale datasets (KonIQ-10K, KADID-10K, FLIVE, and SPAQ), we apply a MultiStepLR schedule that lowers the learning rate by a factor of 0.2 at epochs 15 and 25.

B. Experimental Variance Analysis

Each configuration is trained and evaluated three times with independent random seeds (8, 19, 25). We report the seed-averaged results in the main tables and list the associated standard deviations in Table S2 and Table S4 for cross-dataset experiments.

During these trials, we encountered sporadic numerical instabilities when pairing the CleanDIFT-based SD2.1 backbone with the Sigmoid-first MLP on the SPAQ dataset. In Figure S3 we can notice that the loss explodes for the experiment with the Sigmoid activation in the first hidden layer as it steepens the input distribution, causing many neurons to saturate, the resulting near-zero gradients prevent effective weight updates and precipitate optimisation instability, which is evident in the Figure S3 where the loss remains unchanged. On large-scale datasets such as SPAQ with 40 K samples, the higher number of parameter updates amplifies this vanishing-gradient problem, leading to persistent, high-variance losses and eventual training collapse. We hypothesize that the consistent drop in performance of Sigmoid MLP on larger datasets is due to the same effect; hence, introducing a parallel LeakyReLU branch (our gated activation) restores non-zero gradients, thereby stabilizing training across both small and large data regimes.

B.1. Training Setting

All experiments are run in mixed precision. Model weights are stored in FP16, while the parameters of the learnable activation gates remain in full FP32 to preserve numerical range and prevent gradient underflow. This hybrid setting maintains the speed and memory benefits of half-precision training without compromising the convergence of the gated activations. We train with a physical batch size of 2 and gradient-accumulation of 6, yielding an effective batch size of 12. All experiments were executed on a single NVIDIA A100 GPU. Performance metrics are reported in Table S5

Perception Backbone Building on the observation by Bolya *et al.*[1] that intermediate Perception features can boost downstream performance, we conducted a layer-selection sweep for the **ViT-L/14-336** checkpoint, an ablation not reported in the original paper (see Figure S5). Empirically, features tapped at layer 20 yield the highest validation SRCC on the NR-IQA task, outperforming neighboring layers. Accordingly, all subsequent experiments that use the Perception backbone extract features from layer 20, ensuring a fair and capacity-maximizing comparison with the other encoders.

Dataset Split For every dataset in this study, we perform an 80 / 20 random split training versus validation using the three seed values specified above. The identical protocol is applied to all datasets to ensure consistent evaluation and fair cross-dataset comparisons.

Normalization Unless otherwise noted, we map all opinion scores to the range $[0, 1]$ for a uniform training target. Concretely, we rescale KonIQ-10k MOS by $y/5$ (official MOS are on a 5-point ACR scale), SPAQ MOS by $y/100$ (scores reported on a 0–100 scale), CLIVE MOS by $y/100$ (LIVE Challenge database), FLIVE MOS by $y/100$, AGIQA-3K by $y_{\text{quality}}/5$ and $y_{\text{align}}/5$ (the release provides normalized MOS columns; we standardize to $[0, 1]$ regardless), AGIQA-1K by $y/5$ (normalized MOS in the official spreadsheet), and KADID-10k DMOS by $(y-1)/4$ to convert the $[1, 5]$ range to $[0, 1]$ with higher being better.

C. Embedding Response Analysis

Experiment We use a pretrained SigLIP2 encoder to compute image embeddings and rank feature activations by absolute magnitude. For each percentile band P, we retain features whose magnitudes fall within P and generate Grad-CAM heatmaps on the original image from those features. We visualize four randomly sampled images from each of four datasets, CLIVE, KonIQ10K, KADID10K, and AGIQA1K. CLIVE and KonIQ10K contain authentic distortions; AGIQA1K comprises AI-generated images; KADID10K applies synthetic distortions to natural images. We visualize Top-$N\%$ percentile bands (with $N \in [1, 50]$), letting $S = \{|f_i|\}$, Top-$N\% = \{\, i : |f_i| \geq Q_{1-N/100}(S) \,\}$. See Figures S8–S11.

Observations In our feature attribution analyses, we consistently find that the highest-ranked channels (top decile by response magnitude) correlate most strongly with semantic structure, whereas mid-ranked channels capture more general contextual regularities. The precise band that encodes this "mid-level" context varies across images (e.g., ranks 20–30 for some scenes and 30–40 for others), but the pattern persists. The very top responses align with object and layout level semantics. This helps explain the gains we observe with a sigmoid first-layer activation, by compressing extremes and enlarging sensitivity in the mid-range, sigmoid implicitly regularizes the head to exploit these mid-level features, improving robustness and generalization (Table 4, main paper). Dataset behavior further supports this interpretation. On AGIQA-1K, where degradations are primar-

Table S1. Ablation study comparing frozen backbones and full fine-tuning (FT) against LoRA fine-tuning on the SigLIP2 backbone. The table shows performance gains from allowing backbone adaptation through LoRA (rank=4) versus keeping the backbone frozen. All metrics use "higher is better" scoring. Results are averaged over three runs (seeds: 8, 19, and 25). **Bold** values indicate the better approach between frozen and LoRA for each configuration. Baseline is a three-layer MLP with LReLU activations.

Method	FLIVE		SPAQ		CLIVE		AGIQA3K		KADID10K		KonIQ10K		AGIQA1K		Average	
	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
Baseline (Frozen)	.451	.544	.902	.905	.822	.841	.858	.913	.944	.945	.874	.895	.853	**.891**	.786	.818
Baseline (FT)	.509	.630	.885	.890	.730	.742	.632	.723	.685	.689	.756	.781	.779	.837	.711	.756
Baseline (LoRA)	**.533**	**.641**	**.927**	**.931**	**.875**	**.905**	**.865**	**.917**	**.961**	**.964**	**.932**	**.943**	**.857**	.889	**.844**	**.879**
Baseline (Frozen)+Sig	.537	.620	.896	.899	.827	.852	.861	.916	.913	.911	0.888	0.906	.842	.880	.818	.850
Baseline (FT)+Sig	**.561**	**.659**	**.923**	**.928**	.882	.908	.864	.918	**.963**	**.966**	.931	.946	.845	.880	**.853**	**.886**
Baseline (LoRA)+Sig	.521	.608	.921	.926	**.909**	**.930**	**.878**	**.923**	.939	.943	**.938**	**.947**	**.872**	**.897**	.846	.874

ily generative artifacts that disrupt global semantics (e.g., ill-formed content and structural inconsistencies), quality is tightly coupled to semantic fidelity. Similarly, KADID-10k comprises controlled, intensity-graded distortions; several of these are well captured by strong low-level departures in the feature space, for which LeakyReLU's near-linear pass-through at large magnitudes remains advantageous, consistent with its competitive results on this benchmark. By contrast, on natural-image datasets like CLIVE and KonIQ-10K, where quality information lies in subtle perceptual cues, the sigmoid head excels by prioritizing the mid-to-high semantic regime.

D. Additional Related Work

Recent NR-IQA methods increasingly exploit pretrained vision–language and multimodal models. For example, [7] uses CLIP to assess image quality via prompt pairs with quality-sensitive attributes (e.g., "good photo" vs. "bad photo," or "bright" vs. "dark"). More recent approaches leverage Large Multimodal Models (MLLMs) by replacing scalar labels with text-defined rating levels [8], or by using multi-step reasoning pipelines that first identify the distortion type, then describe its textual impact, and finally compare against a reference image [9]. Other works discretize quality scores through distribution-based formulations [10], or adopt reinforcement learning, such as group relative policy optimization (GRPO), to refine quality-aware policies [3].

References

[1] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025. 2

[2] Honghao Fu, Yufei Wang, Wenhan Yang, Alex C Kot, and Bihan Wen. Dp-iqa: Utilizing diffusion prior for blind image quality assessment in the wild. *arXiv preprint arXiv:2405.19996*, 2024. 1

[3] Weiqi Li, Xuanyu Zhang, Shijie Zhao, Yabin Zhang, Junlin Li, Li Zhang, and Jian Zhang. Q-insight: Understanding image quality via visual reinforcement learning. *arXiv preprint arXiv:2503.22679*, 2025. 3

[4] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019. 1

[5] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7(1):5, 2017. 1

[6] Nick Stracke, Stefan Andreas Baumann, Kolja Bauer, Frank Fundel, and Björn Ommer. Cleandift: Diffusion features without noise. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 117–127, 2025. 1

[7] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2555–2563, 2023. 3

[8] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023. 3

[9] Zhiyuan You, Zheyuan Li, Jinjin Gu, Zhenfei Yin, Tianfan Xue, and Chao Dong. Depicting beyond scores: Advancing image quality assessment through multi-modal language models. In *European Conference on Computer Vision*, pages 259–276. Springer, 2024. 3

[10] Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14483–14494, 2025. 3

Table S2. Standard deviations of SRCC and PLCC across all experiments. Values represent standard deviation across three runs (seeds: 8, 19, 25). This table provides STD values for all experiments mentioned throughout the paper. **Note:** † Abnormally high STD values suggest potential instability in these configurations.

	FLIVE		**SPAQ**		**CLIVE**		**AGIQA3K**		**KADID10K**		**KonIQ10K**		**AGIQA1K**	
Experiment	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
CLIP	.0174	.0002	.0019	.0006	.0245	.0014	.0082	.0019	.0014	.0008	.0056	<.0001	.0081	.0024
CLIP + Sigmoid	.0048	.0001	.0033	.0009	.0174	.0020	.0088	.0016	.0021	.0006	.0009	<.0001	.0088	.0013
CLIP + Activation Gating	.0097	.0002	.0017	.0005	.0182	.0016	.0069	.0020	.0009	.0007	.0024	.0001	.0172	.0018
DINO	.0051	.0003	.0014	.0007	.0436	.0052	.0042	.0019	.0011	.0008	.0136	<.0001	.0112	.0046
DINO + Sigmoid	.0062	.0003	.0007	.0012	.0197	.0060	.0050	.0029	.0082	.0030	.0039	<.0001	.0073	.0036
DINO + Activation Gating	.0054	.0002	.0007	.0010	.0223	.0049	.0099	.0006	.0073	.0012	.0023	.0004	.0103	.0024
DINO-3	.0093	.0002	.0005	.0003	.0410	.0089	.0065	.0013	.0008	.0021	.0019	.0005	.0151	.0017
DINO-3 + Sigmoid	.0029	.0002	.0009	.0012	.0247	.0065	.0110	.0016	.0031	.0015	.0050	.0003	.0128	.0013
DINO-3 + Activation Gating	.0014	.0003	.0013	.0005	.0358	.0051	.0078	.0018	.0022	.0005	.0019	.0005	.0138	.0011
ResNet152	.0290	.0022	.0054	.0020	.0483	.0120	.0316	.0013	.0063	.0047	.0072	.0001	.0349	.0042
ResNet152 + Sigmoid	.0429	.0004	.0013	.0010	.0213	.0139	.0334	.0028	.0234	.0079	.0376	.0001	.0090	.0053
ResNet152 + Activation Gating	.0019	.0008	.0037	.0011	.0454	.0068	.0358	.0028	.0039	.0012	.0226	.0013	.0031	.0041
Perception	.0067	.0003	.0015	.0004	.0167	.0050	.0084	.0026	.0006	.0003	.0211	.0001	.0222	.0024
Perception + Sigmoid	.0065	.0002	.0005	.0003	.0198	.0052	.0095	.0008	.0014	.0007	.0107	<.0001	.0084	.0016
Perception + Activation Gating	.0052	.0001	.0016	.0008	.0150	.0040	.0080	.0004	.0003	.0013	.0031	.0004	.0177	.0030
SigLIP2 (Frozen)	.0076	<.0001	.0024	.0004	.0114	.0007	.0093	.0007	.0008	.0009	.0166	.0001	.0119	.0030
SigLIP2 (Frozen) + Sigmoid	.0037	.0001	.0026	.0004	.0105	.0021	.0120	.0018	.0019	.0016	.0027	<.0001	.0135	.0008
SigLIP2 (FT)	.0065	.0001	.0491	.0099	.1771	.0244	.3099	.0138	.3950	.0398	.2223	.0174	.0977	.0096
SigLIP2 (FT) + Sigmoid	.0009	.0003	.0014	.0003	.0099	.0031	.0054	.0017	.0010	.0002	.0039	.0003	.0126	.0015
SigLIP2	.0416	.0009	.0010	.0006	.0103	.0034	.0116	.0009	.0017	.0007	.0051	<.0001	.0080	.0019
SigLIP2 + Sigmoid	.0047	.0048	.0037	.0006	.0090	.0015	.0092	.0012	.0060	.0015	.0143	<.0001	.0055	.0013
SigLIP2 + Activation Gating	.0224	.0014	.0023	<.0001	.0054	.0026	.0092	.0015	.0014	.0007	.0039	.0003	.0054	.0018
CleanDIFTSD2.1	—	0.013	.0022	3.918†	.0294	.0071	.0097	.0185	.0012	.0003	.0425	.0002	.0059	.0260
CleanDIFTSD2.1 + Sigmoid	.0007	.0014	0.000	43.85†	.0217	.0057	.0121	.0147	.0008	.0006	.0032	<.0001	.0062	.0265
CleanDIFTSD2.1 + Activation Gating	.0040	.0002	.0033	3.915†	.0308	.0066	.0042	.0194	.0009	.0005	.0043	.0003	.0060	.0347

Table S3. Comparison of MLP performance under different feature-retention percentiles k *(dropout variants).* Reported are the mean best Spearman (SRCC) and Pearson (PLCC) for Group 1 (synthetic: AGIQA1K, KADID10K) and Group 2 (natural: CLIVE, KonIQ10K). Δ denotes the change relative to $k = 100$ within each group (more negative indicates a larger drop). Here, k is the retained percentile of features by magnitude used in training/evaluation. *For dropout experiments, the dropout rate is* $(1 - k)$ *(i.e., k is the keep probability).*

Method	k	AGIQA1K		KADID10K		Δ Group 1		CLIVE		KonIQ10K		Δ Group 2	
		SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
MLP-LReLU w/ Dropout Layers	100	.854	.890	.891	.892	0	0	.875	.906	.908	.917	0	0
	90	.855	.890	.883	.884	-0.004	-0.004	.875	.906	.915	.912	-0.001	-0.003
	70	.857	.887	.869	.871	-0.009	-0.011	.880	.907	.896	.896	-0.008	-0.015
	50	.850	.886	.869	.853	-0.013	-0.021	.878	.906	.889	.889	-0.008	-0.014
	30	.826	.868	.865	.861	-0.027	-0.027	.865	.896	.876	.878	-0.021	-0.025
	10	.738	.820	.861	.838	-0.073	-0.062	.812	.812	.864	.876	-0.054	-0.067
MLP-LReLU w/ Input Dropout	100	.855	.891	.885	.888	0	0	.877	.909	.915	.918	0	0
	90	.854	.888	.881	.884	-0.003	-0.004	.876	.905	.913	.914	-0.002	-0.004
	70	.851	.888	.868	.870	-0.011	-0.011	.879	.907	.904	.917	-0.004	-0.003
	50	.852	.885	.864	.865	-0.012	-0.019	.880	.910	.892	.896	-0.010	-0.011
	30	.844	.880	.858	.842	-0.019	-0.028	.882	.909	.891	.892	-0.009	-0.016
	10	.811	.853	.760	.763	-0.084	-0.081	.847	.872	.872	.885	-0.037	-0.047

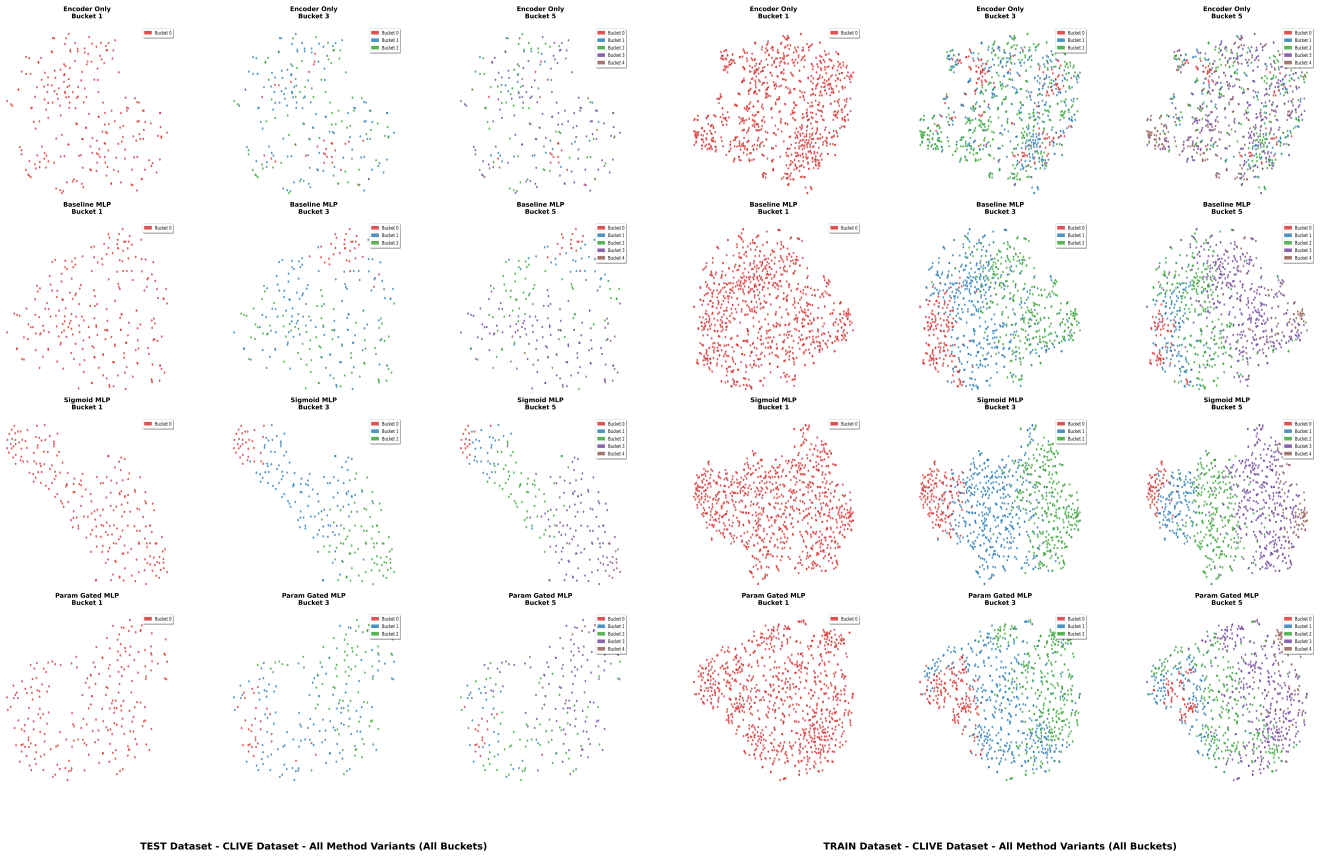

Figure S1. t-SNE visualizations illustrating the contribution of each architectural component. The left panel depicts embeddings from the held-out test split, while the right panel shows the corresponding train-split embeddings. Clear separation across quality buckets in the test plot indicates that the learned representation generalizes beyond the training data.

Table S4. Standard deviations of SRCC and PLCC for cross-dataset evaluations (seeds: 8, 19, 25). Values rounded to 4 decimals.

	Pair 1		**Pair 2**		**Pair 3**		**Pair 4**	
Train	FLIVE		FLIVE		KonIQ10K		CLIVE	
Test	CLIVE		KonIQ10K		CLIVE		KonIQ10K	
Metric	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
SigLIP2	0.0133	0.0226	0.0292	0.0236	0.0071	0.01182	0.0015	0.0021
SigLIP2 + Sigmoid	0.0096	0.0058	0.0015	0.0008	0.0278	0.0236	0.0019	0.0011
SigLIP2 + Activation Gating	0.0249	0.0168	0.0034	0.0071	0.0013	0.0018	0.0050	0.0009

Table S5. Compute profile for SigLIP2 variants. All counts are single-image (batch=1) unless noted.

Variant	Input (H×W)	Patch	MACs (G)	Mem (MiB)	Batch	Precision	Framework	Counter	Attn inc.
SigLIP2 +(LReLU)	512×512	16	*423.99*	*2270*	1	bf16	PyTorch 2.1	fvcore	Yes
SigLIP2 + Sigmoid	512×512	16	*423.99*	*2270*	1	bf16	PyTorch 2.1	fvcore	Yes
SigLIP2 + Activation Gating	512×512	16	*423.66*	*2270*	1	bf16	PyTorch 2.1	fvcore	Yes

Figure S2. t-SNE visualizations illustrating the different activation functions. The left panel depicts embeddings from the held-out test split, while the right panel shows the corresponding train-split embeddings. We observe an interesting phenomenon that Sigmoid activation learns a better representation of the feature space, achieving a better separation.

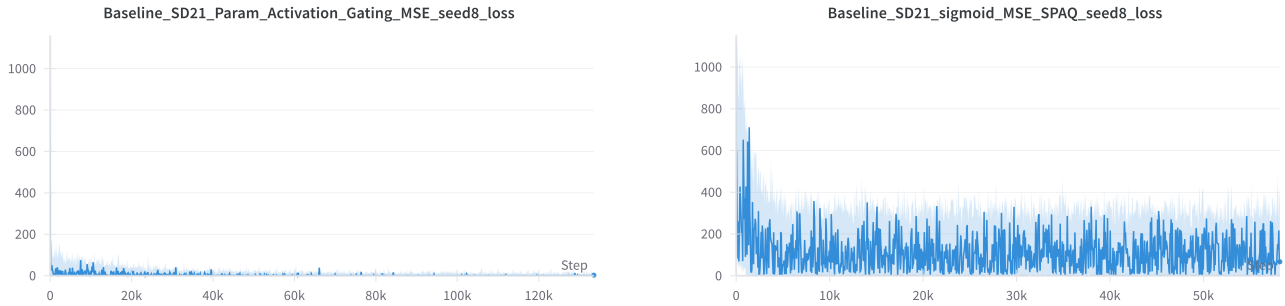

Figure S3. Training loss on SPAQ with the CleanDIFT–SD 2.1 backbone. **Left:** MLP head whose first two hidden layers use LeakyReLU, showing smooth convergence. **Right:** identical MLP but with a Sigmoid in the first hidden layer; the loss spikes and remains unstable, illustrating the saturation-induced optimisation failure discussed in Section B.

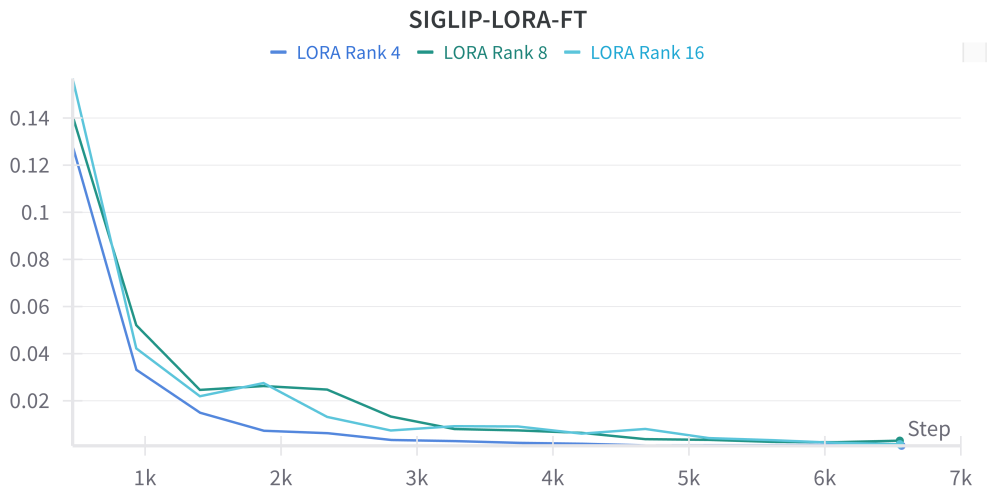

Figure S4. LORA Rank Ablation

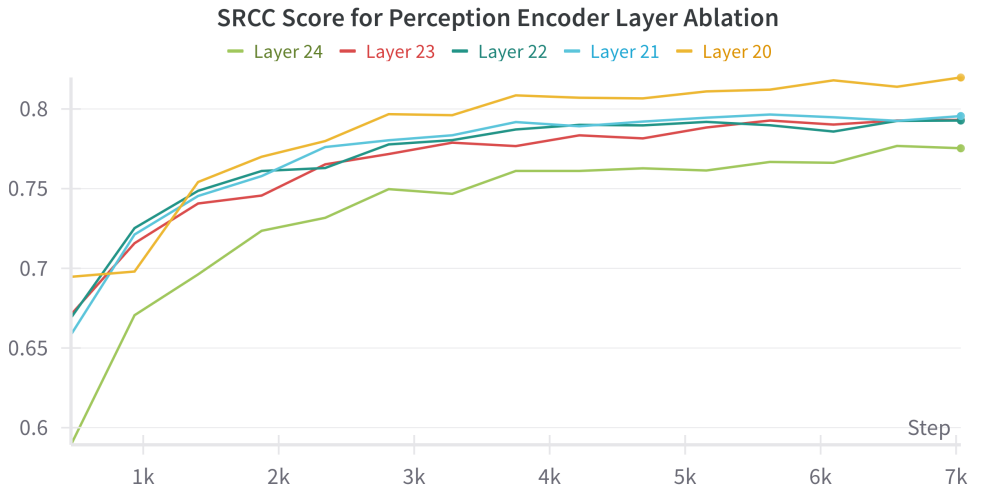

Figure S5. Perception Encoder Layers Ablation (CLIVE Dataset)

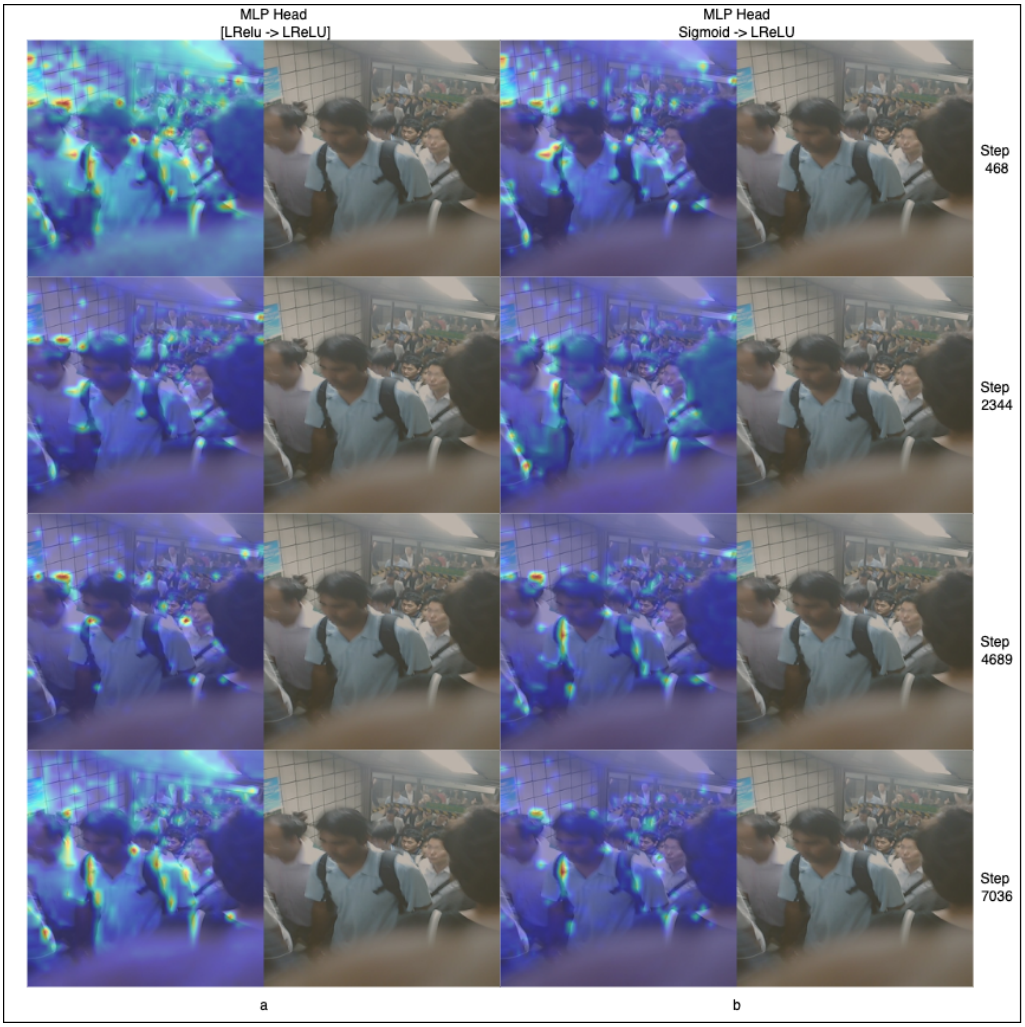

Figure S6. Grad-CAM visualizations comparing an MLP head without sigmoid (a) versus with sigmoid gating (b) over training steps. Introducing the sigmoid reduces the dominance of high-activation (strongly semantic) channels in the backbone features early in training, promoting reliance on medium-response evidence. This yields a clearer correlation with the facial blur artifact in (b), while the baseline in (a) frequently attends to unrelated salient structure and fails to highlight the facial blur.

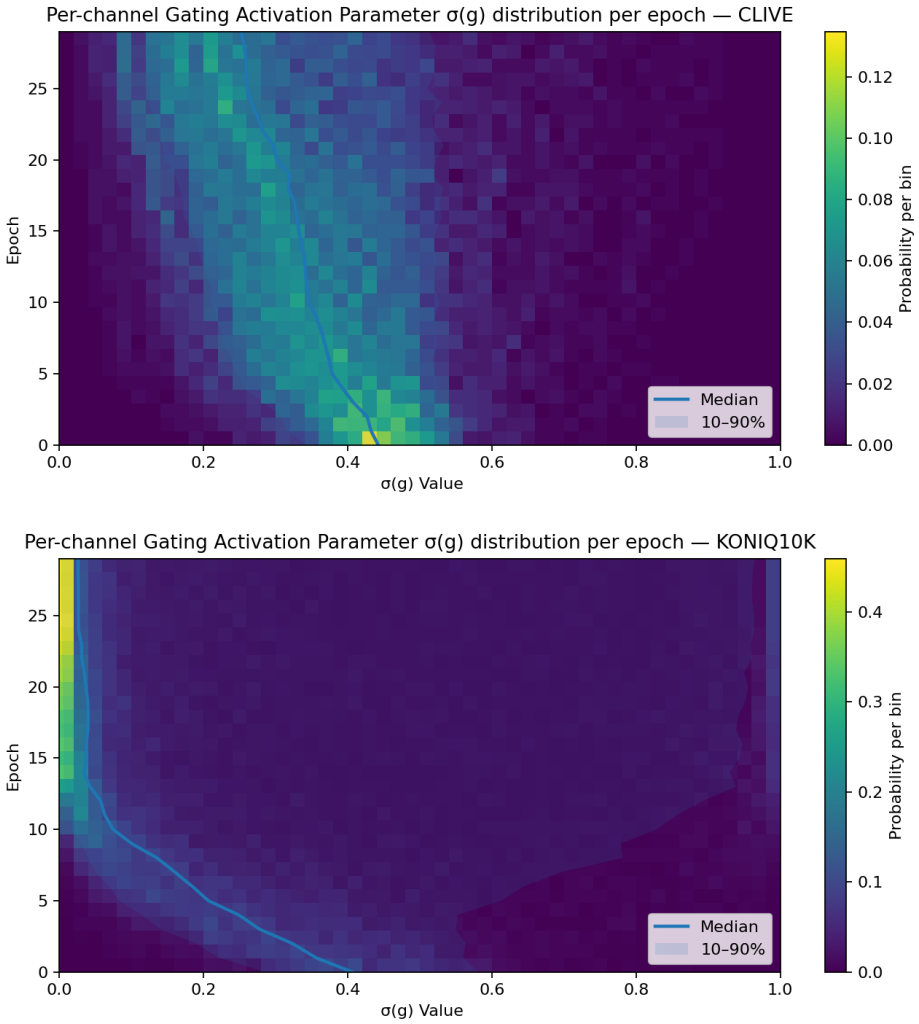

Figure S7. Channel-wise distributions of the gate weight $w = \sigma(g)$learned by the gated activation head across different epochs, comparing CLIVE (low-data regime) and KonIQ-10k (large-data regime). Larger w indicates greater reliance on the sigmoid branch, while smaller w favors the LeakyReLU branch.

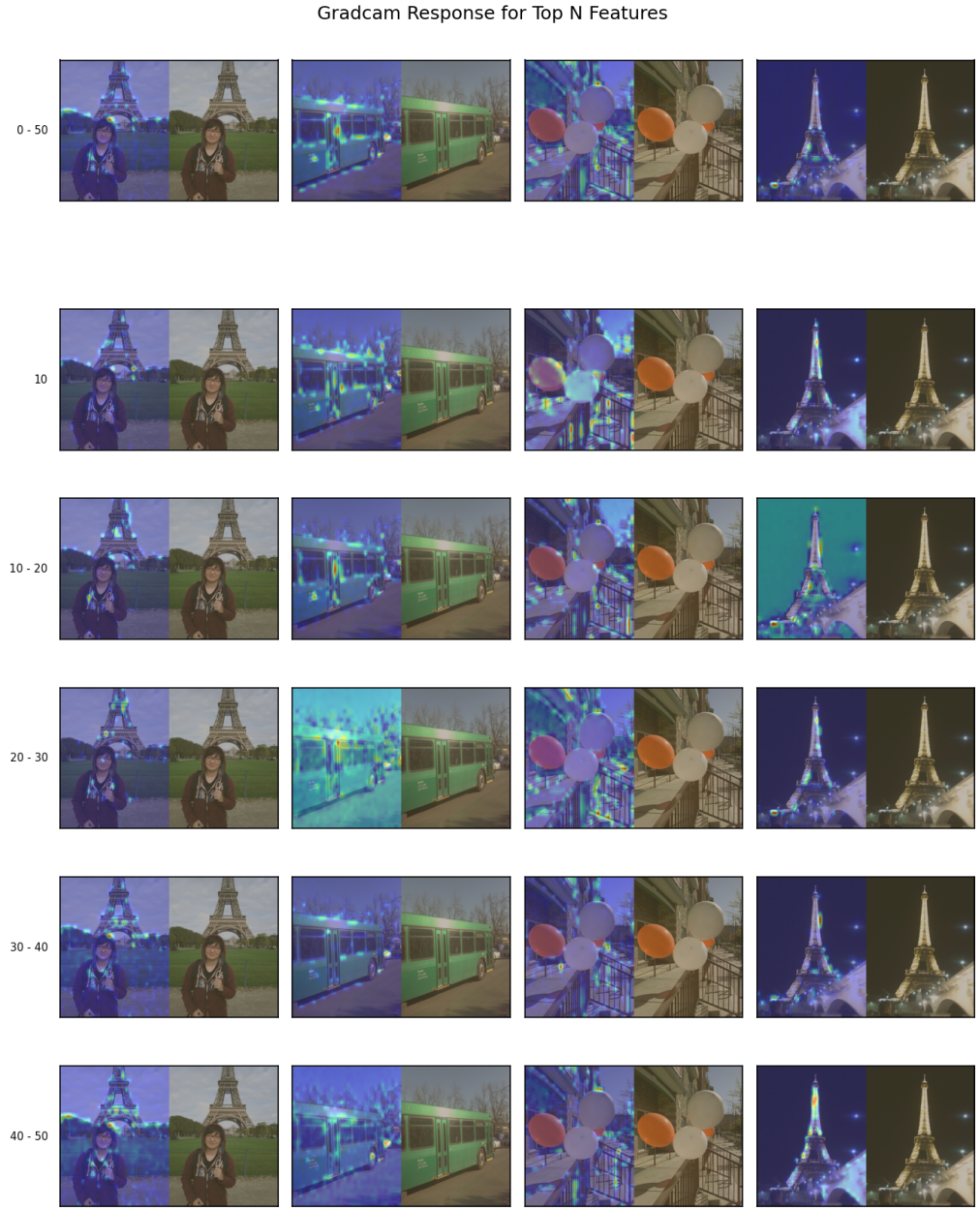

Figure S8. Comparison of Grad-CAM visualizations across SIGLIP2 encoder feature groups on CLIVE images, showing the correspondence between feature responses and input regions. Here, Top-N refers to features whose absolute magnitudes are at or above the Nth percentile.

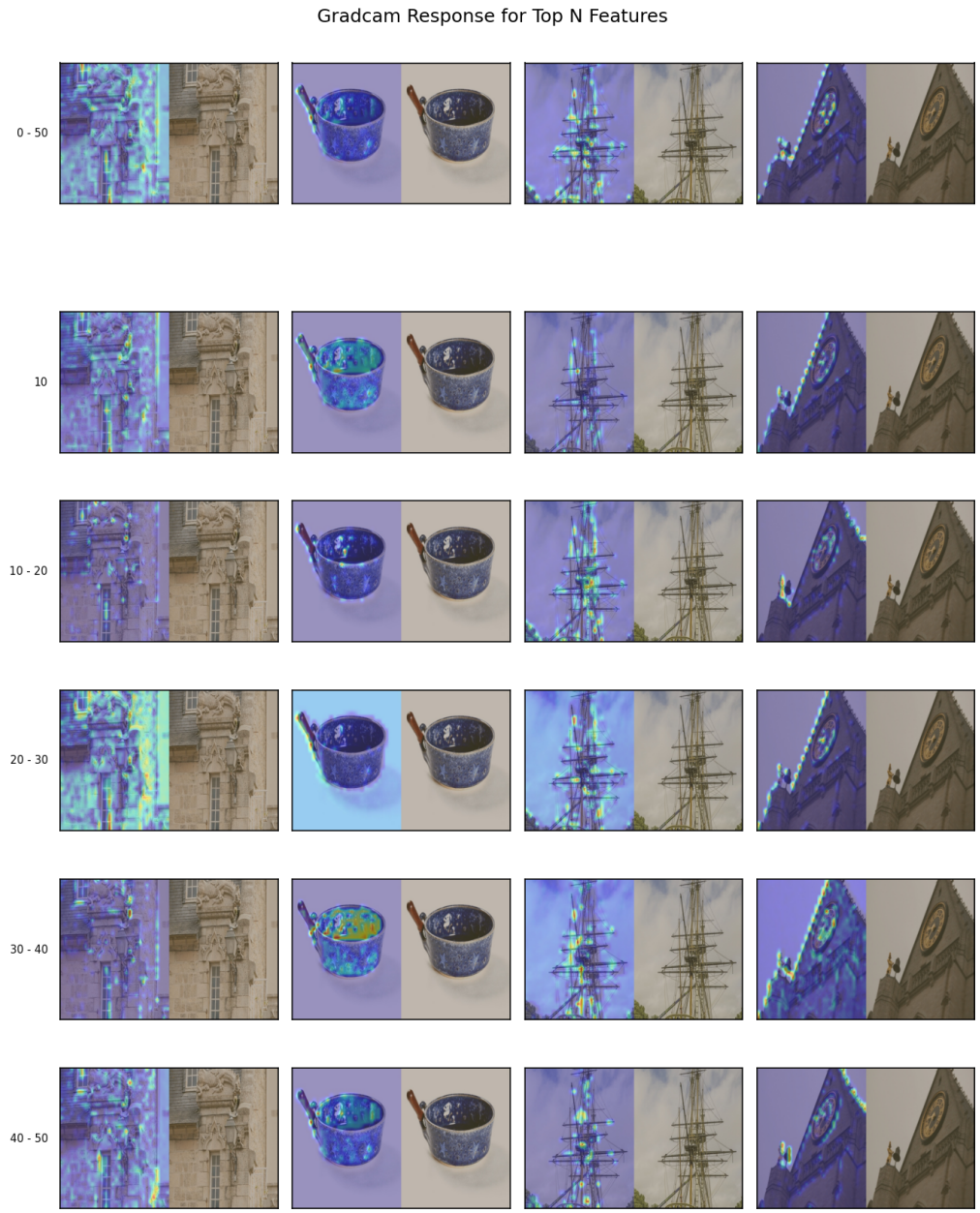

Figure S9. Comparison of Grad-CAM visualizations across SIGLIP2 encoder feature groups on KonIQ-10K images, showing the correspondence between feature responses and input regions. Here, Top-N refers to features whose absolute magnitudes are at or above the Nth percentile.

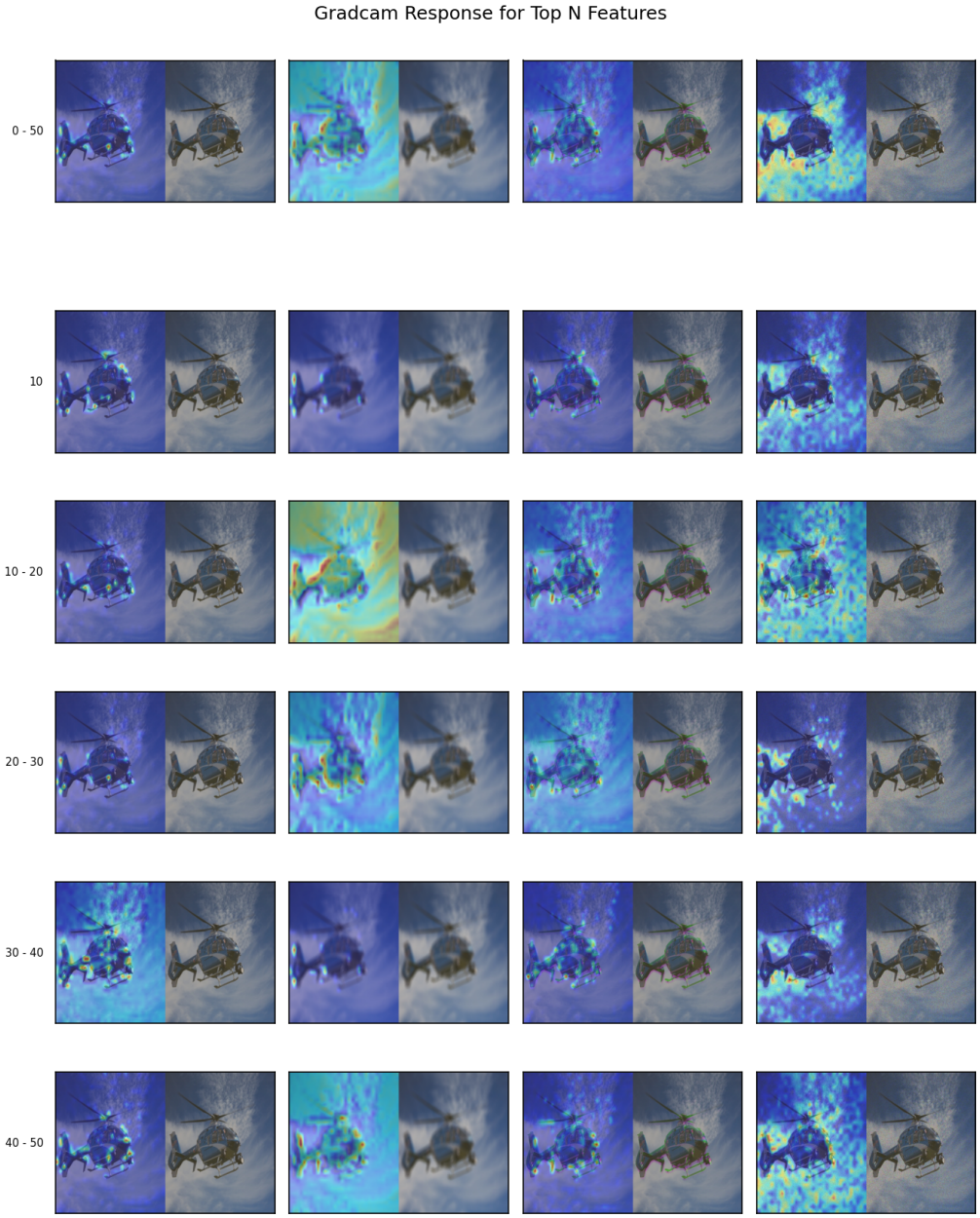

Figure S10. Comparison of Grad-CAM visualizations across SIGLIP2 encoder feature groups on KADID-10K images, showing the correspondence between feature responses and input regions. Here, Top-N refers to features whose absolute magnitudes are at or above the Nth percentile.

Figure S11. Comparison of Grad-CAM visualizations across SIGLIP2 encoder feature groups on AGIQA-1K images, showing the correspondence between feature responses and input regions. Here, Top-N refers to features whose absolute magnitudes are at or above the Nth percentile.