ID-Sim: An Identity-Focused Similarity Metric

Supplementary Material

Appendix Contents

A. Training Data Curation

This section provides technical details of the full curation process: (i) real instance-level data curation decisions such as dataset balancing procedures, criteria used to filter data sources, and more detailed dataset scale information, (ii) generative editing pipeline implementation details for subsets 2a and 2b.

A.1. Subset 1: Real instance-level data

Initial candidate pool for instance-level data. Table 1 lists every dataset originally considered, including several large instance-level datasets not included in the final main training set (e.g., Stanford Online Products [12], MVImgNet [18], Wildlife-ReID [1] datasets). We focus on instance-level datasets available under research-approved licenses that contain more than 500 unique instances. For each dataset we report:

- Total number of images, instances, and categories
- Per-instance image count range
- Type of instance annotation (object ID, catalog ID, animal ID, etc.)
- Known annotation issues (if any)

Dataset	Inst.	Imgs	Cats	Img/Inst	Domain
MET [17]	734	3,429	–	2–17	Art
ILIAS [7]	900	5,326	–	2–35	Everyday Obj
FORB [15]	4,050	16,781	7	2–22	Flat Obj
GLDv2 [13]	4,503	81,964	–	2–6,272	Landmarks
WildlifeReID10k [1]	9,756	126,302	22	1–411	Animals
SOP [12]	11,318	59,551	12	2–12	Products
MVImgNet2* [18]	20,000+	689,003	300+	3–33	Multi-view Obj
DeepFashion [4]	30,018	77,221	13	1–10	Fashion
Total	80k+	–	–	–	

Table 1. **Initial real instance-level dataset pool.** MVImgNet2* is a subset of MVImgNet2 composed of the first two released parts, as the full released dataset contains 180k+ videos. We also use the train_clean split of GLDv2.

Random vs. balanced sampling. Because the number of instances across datasets is highly skewed (e.g., 734 instances in MET vs. 30k+ in DeepFashion2), we first evaluated whether training on a heavily imbalanced mixture would bias the model toward the largest domains. Since ID-Sim is intended to generalize across many visual identities and contexts, we aim to avoid over-representing any specific dataset or domain.

To study the effect of dataset composition, we fix a target pool of 10,000 triplets (30k images) and compare two sampling strategies: (i) proportional sampling based on raw dataset size, and (ii) a balanced sampling strategy designed to equalize per-dataset contribution.

As reported in the main paper, balancing improves the validation ROC AUC from 0.69 to 0.75. We provide the full procedure used to construct the balanced pool below.

Balanced sampling procedure. We sample unique instances rather than individual images to maximize identity diversity. The process is as follows:

1. **Initial allocation.** Each dataset is allocated a quota of $11{,}000/N_{\text{datasets}}$ instances (10,000 for the training set

and 1,000 for validation), giving each dataset an equal starting contribution.
2. **Small-dataset allocation.** Datasets with fewer instances than their quota (e.g., MET, ILIAS) contribute all available instances and are excluded from later steps.
3. **Redistribution.** The remaining instance budget is divided equally among the remaining datasets. This redistribution is repeated until the full 11,000 instance target is reached.
4. **Per-instance sampling.** For each selected instance, we uniformly sample two images from all available images (one anchor, one positive).
5. **Train / validation split.** The datasets are then randomly split into $10,000$ instance train set and $1,000$ instance val set.

After selecting the instances and sampling images, negative pairs are created using *hard-negative mining*: for each anchor image, we search the training pool for the nearest neighbor in DINOv3 embedding space.

This procedure yields two comparable datasets–one proportional and one balanced– whose final instance counts for the training set are:

Dataset	Unbalanced	Balanced
MET	105	663
ILIAS	111	804
FORB	592	1428
GLDv2	625	1419
WildlifeReID10k	1450	1418
StanfordOnlineProducts	1566	1435
MVImgNet2	3191	1411
DeepFashion2	2360	1422
Validation ROC AUC	0.69	**0.75**

Table 2. **Instance counts in the unbalanced vs. balanced 10k training mixtures.**

Other dataset filtering criteria and impact on performance. We observed inconsistencies in how some datasets defined an "instance", especially relative to the definition used in the main paper (shared visual identity). To evaluate whether these inconsistencies affected training quality, we ran an ablation where we applied strict filtering rules to remove ambiguous or overly broad instance labels. Our filtering rules were designed to be simple and reproducible. At a high level, we removed (i) classes where one instance label covered visually different objects, (ii) identities that were extremely difficult to match reliably, and (iii) datasets lacking sufficient contextual variation. Below we describe the exact decisions applied to each dataset.

1. **Incorrect instance granularity.** Several datasets grouped visually distinct objects under the same instance.
   - **FORB**: We removed the `Logo` category because different logo styles (e.g., the "LV" monogram vs. full "Louis Vuitton" text) appeared under one instance label Figure 1.
   - **GLDv2**: Many GLDv2 categories are too broad to represent a single object or a consistent visual identity (see Figure 2). We kept only landmark classes where two random images are likely to show the same physical structure. Specifically, we retained the following hierarchical labels: `[house, lighthouse, tower, skyscraper, observatory, fountain, windmill, sculpture, boat, school, cross, pyramid]`. Broad geographic categories such as `cities`, `mountains`, and `villages` were removed.
   - **SOP**: Product instances in this dataset often included different colors or versions grouped under the same ID. Because these violate our instance definition, we removed SOP entirely.
2. **Hard-to-match or viewpoint-inconsistent identities.** Some identities were not mislabeled but were visually too difficult for consistent matching, either due to limited texture cues or extreme viewpoint differences.
   - **WildlifeReID10k**: Certain animal identities (e.g., belugas, dolphins) in this dataset have little to no distinctive patterning and appear nearly indistinguishable across individuals. Others include opposite-sided views of the same animal under the same identity, making consistent matching unreliable. To avoid these failure cases, we retained only the DogFaceNet and CatIndividualID subsets, which have stable markings and consistent viewpoints.
3. **Insufficient contextual variation.**
   - **MVImgNet**: Although MVImgNet provides rich multi-view rotation, it contains very limited background or lighting variation within a single instance sequence since it is a multi-view dataset. Because our training objective requires seeing the same instance under diverse contexts, we removed MVImgNet for insufficient contextual diversity.

These filtering steps remove ambiguous labels and ensure that the remaining datasets align with our visual instance definition. After this filtering, we perform the balancing again following the procedure outlined in Section A.1. Together with the sampling, this process produces a cleaner and more consistent dataset, significantly boosting evaluation validation performance from 0.75 to 0.89 (see Table 3).

A.2. Subset 2: Synthetic data

While the real instance-level datasets in Subset 1 provide strong coverage of identity-preserving variation, they un-

Figure 1. **Filtered out FORB `logo` category.** We observe consistent appearance inconsistencies between the same "instance" category in FORB's "logo" class.

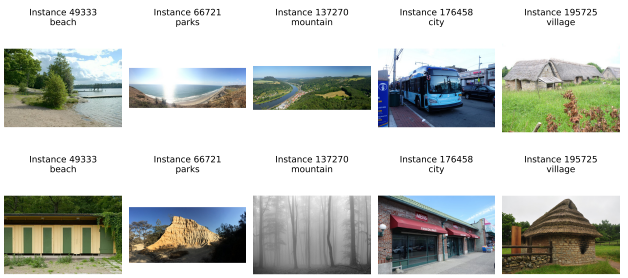

Figure 2. **Filtered GLDv2 categories.** Many GLDv2 classes cover broad geographic areas rather than a single localized site, building or an object, making it difficult for a class to correspond to a consistent visual identity.

Dataset	Filtered
MET	671
ILIAS	826
WildlifeReID10k (Dogs and Cats)	1501
FORB (Filtered)	2346
GLDv2 (Filtered)	2315
DeepFashion2	2341
Validation ROC AUC	**0.89**

Table 3. **Instance counts for the filtered and balanced 10k training mixture.**

derrepresent many forms of contextual change (e.g., background, lighting, scene geometry). To address this limitation, we augment our training set with synthetic data generated through controlled editing. These edits preserve the object's visual identity while introducing new contexts that rarely appear in the original datasets.

We use two complementary sources of synthetic data. **Subset 2a** applies generative contextual edits (background and lighting changes) to isolated frames sampled from video datasets. **Subset 2b** applies generative foreground edits to create hard-negative examples for contrastive triplets.

A.2.1. Subset 2a: Contextual Edits for Generative Synthetic Positives

Base datasets. Subset 2a is constructed from sequential video datasets rather than independent images of an instance. Video sources are particularly valuable because they capture the same instance under natural pose variation but often lack diversity in background and illumination. Generative editing conditioned on these real frames therefore injects controlled contextual diversity while maintaining identity fidelity. We use LaSOT [3], GOT10k [5], YouTube-VIS [16], and UCO3D [9], chosen for their instance diversity, scale, and availability of mask annotations. To obtain a diverse and high-quality set of frames per instance, we use a simple dataset-adaptive sampling strategy:

- **Long videos (>6s):** divide each sequence into 5–6 equal-duration parts and sample one valid frame from each part
- **Short videos (≤6s):** sample every $k_{\text{annotated}} \times$ annotation_stride frames

Dataset-specific values for frame rate, annotation stride, window size, and number of segments are provided in Section A.2.1. These parameters are chosen so that each sampling window corresponds to roughly 1–2 seconds of video, preventing oversampling of near-duplicate frames.

Within each sampling window, we apply the following quality filters:

- **Foreground coverage:** between 10% and 90%
- **Sharpness:** blur score [11]. >50, where the blur score is computed as the variance of the Laplacian (higher variance indicates a sharper image with stronger edges)

If multiple frames satisfy these criteria, we randomly select one to encourage temporal diversity. Instances are retained only if at least two valid frames are obtained.

Dataset	Frames per Second (FPS)	Frame Stride	Number of Frames (k)	Window Size	Number of Sampled Parts
LaSOT	30	5	6	30 (1s)	6
GOT-10k	10	5	2	10 (1s)	5
YouTubeVIS	6	5	2	10 (1.7s)	5
UCO3D	30	1	30	30 (1s)	5

Table 4. **Dataset-specific parameters for video frame sampling.** About 5-6 frames are sampled from each instance sequence, at intervals that are at ∼1 second apart.

Editing model and pipeline. Contextual edits are produced using the Qwen-Image-Edit [14] diffusion model (`Qwen/Qwen-Image-Edit`) with Lightning LoRA weights, enabling 8-step inference. All generations use bfloat16 precision and a `FlowMatchEulerDiscreteScheduler`. A fixed generator seed ensures determinism.

Preprocessing. Each selected frame is paired with its binary foreground mask. Before editing, we apply:

- Foreground crop preserving a 2:3 or 3:2 aspect ratio.
- Resize so the longer side is at most 1248 px and both dimensions are divisible by 32.
- Foreground scaling: if the mask covers less than 10%, scale up to exactly 10%; otherwise randomly select a scale factor so the new coverage lies between 10% and the original value. Each instance is assigned a scale mode (small or large).
- Placement of the scaled foreground onto a white canvas without border clipping.
- Composition of the foreground onto the blank canvas to form the editing input.

Prompts. Each frame receives a unique background–lighting combination. Background prompts are sampled from a supercategory-specific list using `category_to_supercat.json` and `supercat_to_backgrounds.json`. Lighting prompts come from `lighting_prompts.json` and are conditioned on the selected background using `background_to_scene.json`. The prompts are generated using GPT-4o [6]. All component files are included in the supplemental.

The prompt used for all contextual edits is:

```
Replace only the white background
pixels with {background}; keep
the foreground objects and text
completely unchanged in size,
position, orientation, and
appearance (except lighting);
preserve original text, composition,
proportions, alignment, and text
properties; seamlessly blend the
new background with simulated
{lighting_prompt} to match scene
lighting, shadows, and reflections;
ensure natural integration without
duplication, movement, or distortion
of the foreground; maintain original
dimensions, aspect ratio, and focal
center; adjust foreground lighting
for seamless blending.
```

Parameter sampling. Contextual edits use fixed settings: 8 inference steps, guidance scale of 1.0, and a fixed generator seed. Background and lighting indices are sampled per frame.

Generation and output. The model replaces only white-background pixels with the selected scene while blending foreground lighting to match. Outputs include the edited RGB image and updated mask, saved with deterministic filenames encoding all edit parameters.

Use in training. During training, triplets are formed by mixing original and edited views of the same instance. Each positive pair is chosen uniformly from:

1. original anchor + edited positive
2. edited anchor + original positive
3. edited anchor + edited positive

The negative is drawn from either an edited or original view of a different instance. Examples are shown in Figure 3. Adding this dataset in a 1:1 mix with Subset 1 results in a **validation ROC AUC improvement from 0.89 to 0.937**, as reported in the main paper, suggesting that the diversification of contextual edits helps. The ablation on the dataset composition is in Section B.1.

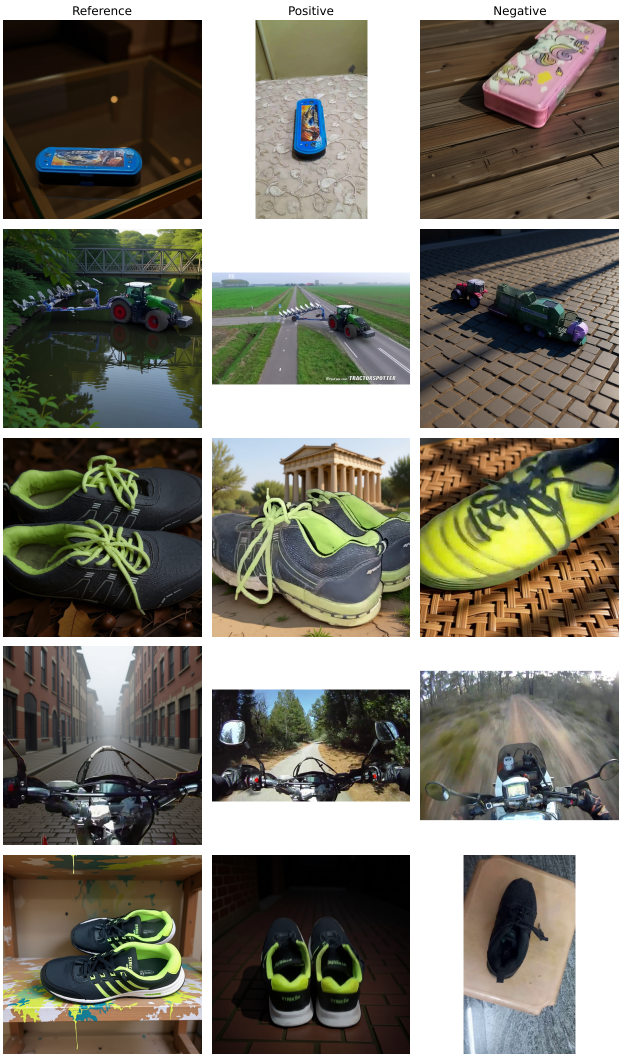

Figure 3. **Generative Contextual Edited Images**

A.2.2. Subset 2b: Identity-Altering Edits for Hard Negatives

Subset 2b introduces controlled foreground edits that alter identity-defining features while maintaining class-level semantics. These edits produce realistic but non-matching variants of each object and are used exclusively as hard negatives.

Base datasets. We apply identity edits only to datasets with high-quality segmentation masks: DeepFashion2 [4] and UCO3D [9]. These datasets provide clean boundaries and stable viewpoints. The remaining video datasets contain motion blur or noisy masks, and the real instance-level datasets from Subset 1 do not provide per-object masks.

Model and pipeline. Foreground edits are generated using the `FluxFillPipeline` (`black-forest-labs/FLUX.1-Fill-dev`) [8]. The model operates in bfloat16 precision and performs inpainting only where the input mask is white. The background is preserved unchanged.

Preprocessing. All images and masks are resized so the longer side is at most 720 px, while preserving aspect ratio and ensuring both dimensions are divisible by eight. We apply randomized partial masking by removing 40–60% of the foreground region along a horizontal or vertical axis to introduce occlusion and increase edit diversity.

Prompts. All identity edits use a class-specific prompt of the form:

```
Photo of a <object>
```

The object name is drawn from `instance_class_to_name.json`. These prompts encourage edits that remain faithful to class semantics while modifying fine-grained appearance cues.

Parameter sampling. For each frame we sample:

- Strength in the range 0.5–0.8
- 50 inference steps
- Guidance scale of 2.5

Sampling is controlled by a per-frame seed computed as `seed + row_id`. The generation step uses a fixed internal seed of zero, enabling fully deterministic output.

Generation and output. The pipeline inpaints only the masked foreground, producing realistic but identity-altered variants. Outputs are resized back to the original crop resolution and saved with deterministic filenames encoding all parameters.

Figure 4. **Generative Edited Hard-Negatives**

Use in training. Identity-edited images are used only as hard negatives. To avoid the model relying on generative artifacts to identify these negatives, we add mild generative noise (strength 0.1) to the anchor and positive whenever a triplet includes an identity-edited negative. This noise does not change image content but prevents artifact-based shortcuts. Negatives are sampled from both edited and original images of other instances. Examples appear in Figure 4. Adding this dataset in a 1:1:1 mix with Subset 1 results in a **validation ROC AUC improvement from 0.937 to 0.965**, as reported in the main paper, suggesting that the edited hard-negatives provide additional signal. The ablation on the dataset composition is in the following section.

B. Ablation Studies

B.1. Ablation Datasets

For model selection we rely on two complementary validation sets. First, we use the validation split of the curated real instance dataset from Subset 1, which we refer to as **Real Instance Validation**. The metric on this set is accuracy as it is in a triplet format. Second, we construct a small identity-focused validation set composed of five Flux-generated base instances, each edited with a mixture of identity-preserving and identity-altering Photoshop [2] modifications. We evaluate binary identity classification on this set and refer to the resulting score as **Identity Validation**.

This identity-focused set is intentionally small, does not overlap with any test data, and provides an additional targeted signal that complements the broader Real Instance Validation set. Throughout the ablation experiments, we report results on both validation sets and use them jointly to identify the best-performing configuration.

B.2. Training Dataset Ablation.

To determine the appropriate training scale and composition, we evaluate both factors using two complementary validation metrics. As shown in Table 5, performance increases gradually when scaling from 5k to 20k instances, but the gains are modest and within the range of expected variance. Beyond 20k, identity-validation performance decreases substantially. This pattern suggests that the main benefits are already achieved at moderate dataset sizes, and that 10k instances provide a stable and efficient operating point.

Dataset Scale	Real Instance Validation (Acc)	Identity Validation (ROC AUC)
5k even split	0.860	0.97
10k even split	0.870	0.97
20k even split	0.880	**0.98**
30k	**0.8825**	0.91

Table 5. **Effect of training dataset scale on validation performance.**

Next, we evaluate dataset composition under a fixed 10k training set, comparing an even (1:1:1) mixture to configurations where one subset is made dominant (0.7:0.15:0.15). As shown in Table 6, the even split achieves the best balance across both validation metrics, whereas skewed mixtures improve one metric at the expense of the other.

Based on these findings, we adopt the **10k even-split** configuration as our final training mixture, providing strong and stable performance across real-instance and identity-level evaluations.

Dataset Composition	Real Instance Validation (Acc)	Identity Validation (ROC AUC)
Even Split	0.8715	**0.97**
Subset 1 dominant	**0.8905**	0.95
Subset 2a dominant	0.8715	0.95
Subset 2b dominant	0.8685	0.96

Table 6. **Effect of dataset composition on validation performance (fixed 10k scale).**

B.3. Training Ablation

All ablations in this section are evaluated using a **joint CLS+patch embedding** and are trained using the 10k balanced training mixture.

We conduct ablations to isolate the contribution of the backbone, feature losses, and patch similarity metrics. These models are evaluated on two validation sets: (i) Real Instance Validation (accuracy) and (ii) Identity Validation (ROC AUC).

B.3.1. Backbone and Input Resolution

Backbone / Resolution	Real Instance Validation (Acc)	Identity Validation (ROC AUC)
DINOv3-L/16 @ 448 (Baseline)	**0.8715**	**0.965**
DINOv3-L/16 @ 224	**0.8715**	**0.965**
DINOv3-B/16 @ 448	0.834	0.921
DINOv2-L/16 @ 448	0.8355	0.895
DINOv2-L/16 @ 224	0.8135	0.819

Table 7. **Backbone and resolution ablation.**

DINOv3 performs better than DINOv2 across both validation sets, and resolution mainly affects DINOv2, but higher resolution results in better performance. ViT-L architecture outperforms ViT-B. This supports using **DINOv3-L/16 at 448px** in the final model.

B.3.2. CLS vs. Patch vs. Joint Training

Feature Loss Setting	Real Instance Validation (Acc)	Identity Validation (ROC AUC)
CLS + Patch (Baseline)	**0.8715**	**0.965**
Patch Loss Only	0.8665	0.908
CLS Loss Only	0.8065	0.893

Table 8. **Ablation on CLS vs. patch vs. joint training.**

Joint supervision combines the complementary strengths of CLS and patch features, resulting in stronger overall performance than using either feature in isolation.

B.3.3. Loss Function and Patch Metric

Sinkhorn OT improves patch alignment over cosine distance, and InfoNCE provides the strongest identity separa-

Objective / Patch Metric	Real Instance Validation (Acc)	Identity Validation (ROC AUC)
InfoNCE + Sinkhorn (Baseline)	**0.8715**	**0.965**
InfoNCE + Cosine Patch Metric	0.8655	0.940
Hinge + Sinkhorn	0.8705	0.945
BCE + Sinkhorn	0.8395	0.923

Table 9. **Loss and patch similarity ablation.**

tion among the tested objectives. These results support the choice of **InfoNCE with Sinkhorn** in the final model.

B.3.4. Overall Summary

Across all ablations, the combination of DINOv3-L/16, joint CLS+patch training, and Sinkhorn OT produces the most reliable identity-sensitive behavior and is therefore adopted in all main-paper experiments.

C. Training Details

With the architecture and dataset design fixed as above, we ablated over the key training hyperparameters and arrived at the following final configuration.

C.1. Model Configuration

- **Backbone**: DINOv3-ViT-L/16 (stride 16), using CLS and patch features
- **Head**: dual MLPs with 512-dim hidden layers (CLS and patch)
- **LoRA adaptation**: rank 16, $\alpha = 32$, dropout 0.05
- **Input resolution**: 448×448
- **Precision**: `bfloat16`

C.2. Optimization

- **Optimizer**: AdamW, learning rate 3×10^{-4}, weight decay 0
- **Batch size**: 8 (effective batch size 32 with $\times 4$ gradient accumulation)
- **# epochs**: 3

C.3. Loss

- **Objective**: InfoNCE with single-negative sampling
- **Margin**: 0.1
- **Feature weighting**: CLS : Patch = 1 : 1
- **Patch alignment**: Sinkhorn optimal transport

C.4. Data Augmentations

- Random resized crop (scale 0.9–1.0; aspect ratio 1:1; bicubic)
- Color jitter (brightness 0.2, contrast 0.2, saturation 0.08, probability 0.8)
- Gaussian blur (kernel 7×7, $\sigma \in [0.05, 0.6]$, probability 0.5)

C.5. Sinkhorn Patch Metric

- **Implementation**: `geomloss.SamplesLoss` with $p = 2$
- **Regularization**: 0.05
- **Blur**: 0.05
- **Maximum tokens**: 1024
- Patch features L2-normalized before distance computation

C.6. Data Loading

- 4 dataloader workers
- Up to 3 concurrent S3 downloads
- Train/val splits loaded from S3 parquet files

D. Evaluation

D.1. Evaluation Dataset Details

We summarize here the seven evaluation datasets used in the main paper. Each dataset follows its standard evaluation protocol and is fully disjoint from the training data. The only exception is DeepFashion2, for which a subset of the dataset is used during training; however, the evaluation split employed here is strictly non-overlapping with the training split.

PODS (Instance Retrieval)

- **Task:** Instance retrieval for personalized household objects (Figure 5).
- **Size:** 1,200 query images and 300 gallery images.
- **Instances:** 100 object instances appearing in both splits.
- **Labels:** Instance-level ID.
- **Protocol:** We evaluate using the dataset's canonical setup: the 1,200 images from the `test_dense` split serve as queries, and the 300 images from the `train` split serve as the gallery. Although the split is named "train," it is only part of the dataset organization and is not used to train our model.
- **Metrics:** mAP (main), ROC-AUC, nDCG. Normalized Discounted Cumulative Gain (nDCG) evaluates how well the ranking prioritizes the most relevant matches.
- **Notes:** Images show controlled variation in viewpoint, background, and lighting across all instances.

DeepFashion2 (Instance Retrieval)

- **Task:** Clothing-item instance matching across domains (Figure 6).
- **Size:** 1,668 queries; 3,065 gallery images
- **Instances:** 1,668 clothing items
- **Labels:** Per-item instance ID
- **Protocol:** Standard fashion retrieval (each query has at least one gallery match)

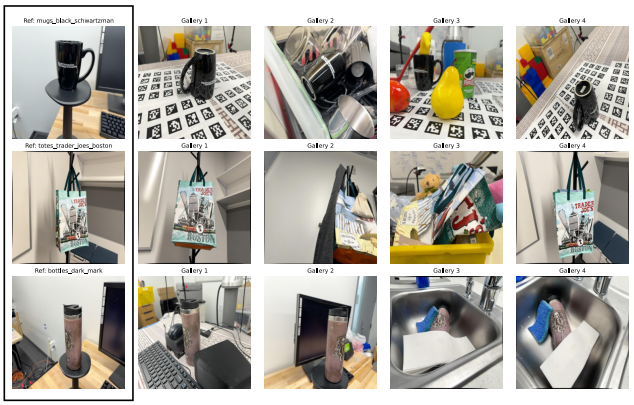

Figure 5. **PODS Dataset.** The dataset is composed of household objects occurring under different distribution shifts, with varying backgrounds, distractor objects, and poses.

- **Metrics:** mAP (main), ROC-AUC

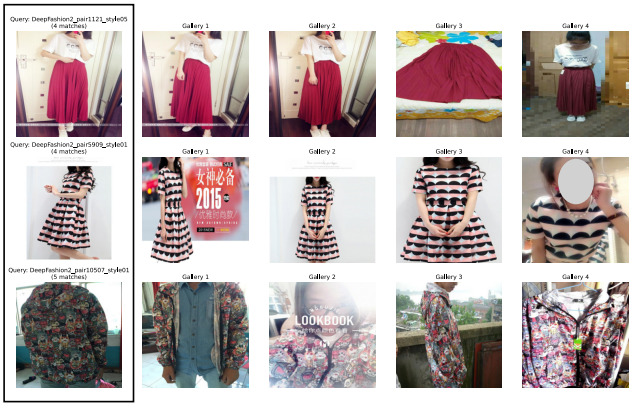

Figure 6. **DeepFashion2.** The DeepFashion2 dataset features query / gallery images of the same clothing item in-shop as well as worn by consumers.

AerialCattle2017 (Animal Re-ID)

- **Task:** Animal identity retrieval from aerial imagery (Figure 7)
- **Size:** 2,329 filtered images
- **Identities:** 23 cattle
- **Splits:** 23 queries; 2,306 gallery images
- **Labels:** Individual animal ID
- **Protocol:** Rank gallery images for each query
- **Metrics:** mAP (main).

PetFace (Animal Re-ID - Verification)

- **Task:** Pairwise identity verification across 13 species
- **Size:** 3,250 pairs
- **Labels:** 1,622 positive; 1,628 negative
- **Species:** 13 unique species: cat, chimp, chinchilla, degus, dog, ferret, guineapig, hamster, hedgehog, javasparrow, parakeet, pig, rabbit

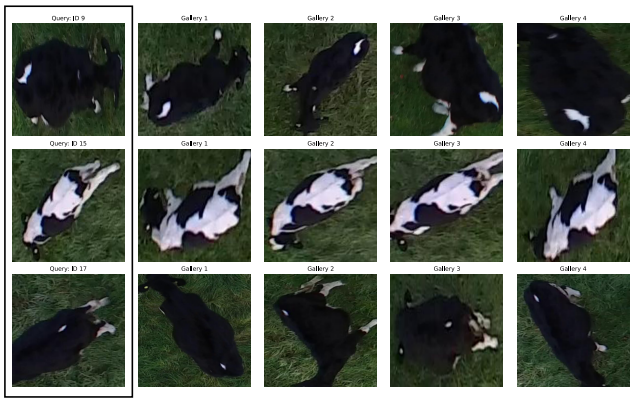

Figure 7. **AerialCattle2017.** This dataset is composed of aerial imagery of various cows on fields, and the task is to retrieve the same individuals based on a query image.

- **Protocol:** Predict whether two images depict the same individual
- **Metrics:** mAP (main)

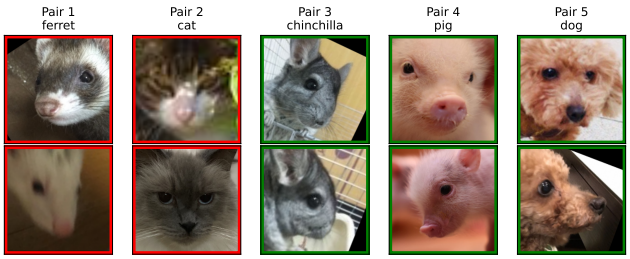

Figure 8. **Petface.** Evaluation benchmark of 13 unseen animals. Red depicts different individual and green depicts same individual.

CUTE (Triplet Matching)

- **Task:** Fine-grained object discrimination using triplet matching
- **Size:** 1,800 triplets
- **Structure:** Each sample contains an anchor, a positive (same instance), and a negative (different instance)
- **Modes:** 1) *Easy* mode uses triplets in which all three images come from the same scene, testing discrimination between instances under identical background and context, 2) *Hard* mode selects the anchor from a different scene whenever possible while keeping the positive and negative in the same scene; this requires recognizing the same instance across scene changes while rejecting a same-scene negative
- **Protocol:** Predict whether the anchor is more similar to the positive than the negative
- **Metrics:** Accuracy (main). We report **Hard-mode results** in the main paper and provide both modes in the supplemental

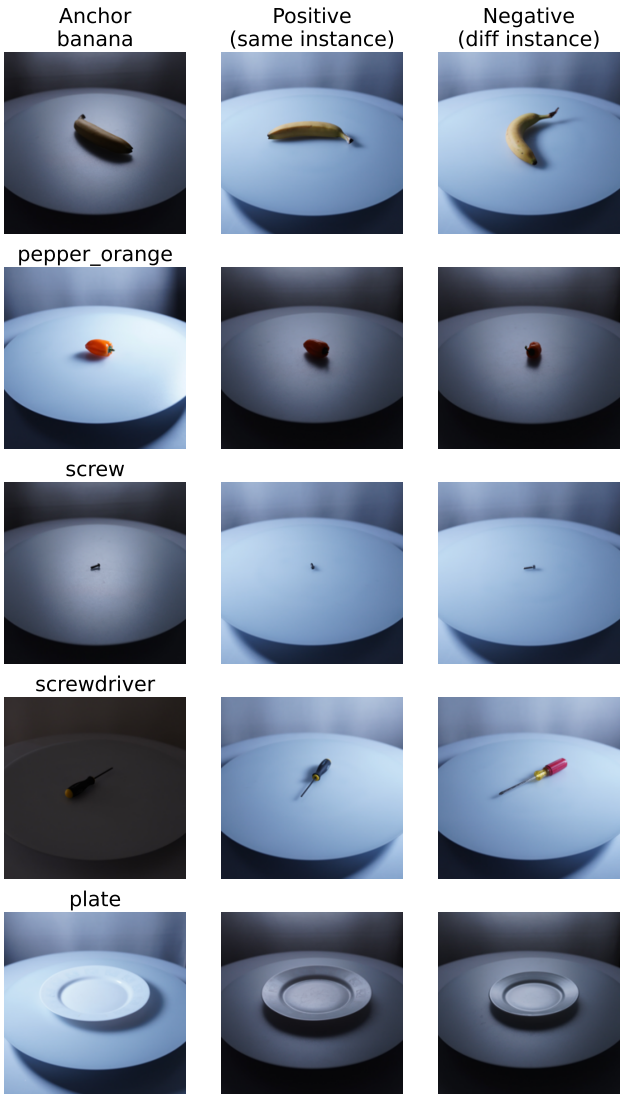

Figure 9. **CUTE triplets.** Examples of triplets selected for "Hard" mode. The positive and negative examples are drawn from the same scene type, which may be different from the scene type of the anchor, forcing a match across extrinsic characteristics.

Subjects2k (Binary Verification for Generative Model)

- **Task:** Human-validated concept preservation
- **Size:** 2,000 pairs
- **Labels:** 473 positive; 1,527 negative
- **Source:** Curated from Subjects200k using GPT-4V filtering + human annotation
- **Protocol:** Predict whether the target preserves the identity of the reference
- **Metrics:** AP (main), ROC-AUC

Figure 10. **Subjects2k Pairs.** Newly annotated 2k subset of Subjects200k [19]. Green depicts same instance, red is different.

DreamBench++ (Discrete Rating for Generative Model)

- **Task:** Identity preservation in subject-driven image generation
- **Size:** 6,921 valid pairs (after filtering)
- **Ratings:** Discrete identity score in $[0, 4]$
- **References:** 110 reference subjects
- **Protocol:** Rank generated images by predicted similarity to the reference
- **Metrics:** Spearman correlation (main), Kendall correlation

Figure 11. **DreamBench Pairs.** DreamBench images are accompanied by human annotations out of 4.

D.2. Subjects2k Human Annotation Pipeline

Motivation DreamBench++ [10] is one of the most widely used human benchmark for evaluating concept preservation in personalized generation, but its annotation design introduces significant noise. Each image receives only two human ratings, and annotators provide a 0–4 rubric score rather than answering a direct same/different or pairwise comparison questions. As shown in Figure 18, this results in both (i) large identity variation among images with identical DreamBench scores, and (ii) highly variable scores for images with higher identity similarity. These inconsistencies motivate the need for a cleaner, better-calibrated evaluation set. We therefore construct **Subjects2k**, a new human-annotated subset of Subjects200k [19] designed to provide more reliable identity-preservation judgments.

Subjects2k: Setup. Subjects2k is derived from the GPT-annotated [6], Flux-generated [8] Subjects200k [19] dataset used for high-fidelity image editing evaluation. Subjects200k provides a 0–5 score per image indicating GPT's assessment of identity preservation. From this pool, we construct a balanced human-evaluation subset by sampling 1,000 images with GPT score 5, and 200 images from each of the remaining scores 0-4, yielding 2,000 images total. We built a lightweight web interface (custom server shown below) and collected human annotations on Prolific.

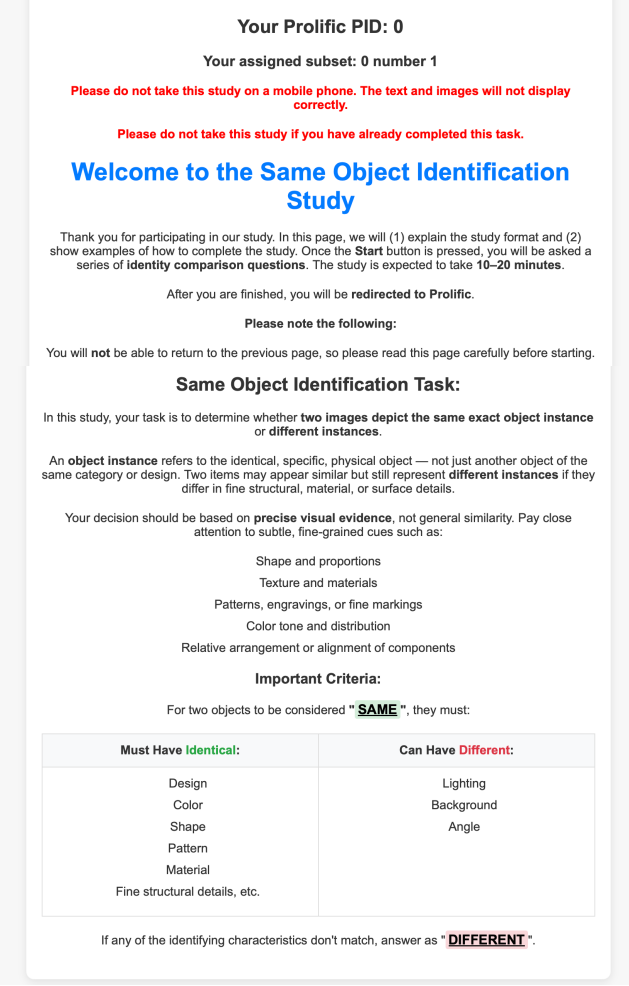

Figure 12. **Introduction Page for Subjects2k Annotation Server.** We provide a clear definition of an instance to all participants prior to starting their annotations.

Subjects2k: Human Annotation Summary We collected human judgments for all 2,000 image pairs in Subjects2k and inserted 7 manually-verified sentinel pairs with known ground-truth labels. After each annotation batch, we filtered annotators by requiring perfect accuracy on all sentinel questions; responses from any annotator who missed

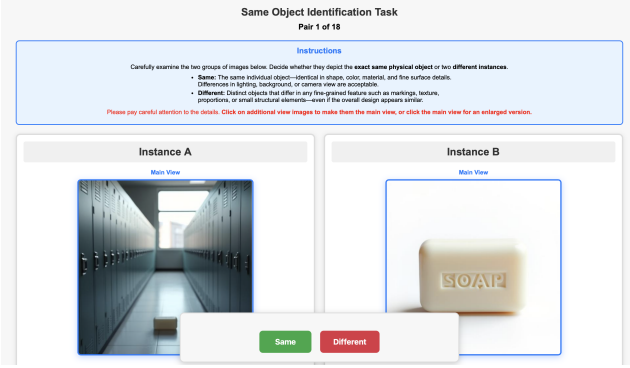

Figure 13. **Subjects2k Annotation Server Task Page.** Example of a task page for our annotators.

one or more sentinels were discarded. This procedure ensured a high-quality, reliable annotation set. Each pair was annotated in batches: we first obtained labels from three annotators (post-filtering). If all three agreed (all "same" or all "different"), we stopped for that pair. If there was any disagreement, we collected up to five additional annotations, for a maximum of nine annotations per pair. This procedure yields an average of 5.01 annotations per pair (min. 3, max. 9).

Agreement measure. For each pair, let p be the fraction of annotators voting "same instance". We define agreement as $\max(p, 1-p)$, i.e., the fraction of annotators supporting the majority label. Averaged over all 2,000 pairs, the agreement is 0.864.

Continuous labels and binarization. For each pair, we define a continuous label

$$\ell = \frac{\#\text{"same" votes}}{\#\text{total votes}} \in [0, 1].$$

The empirical distribution of ℓ is summarized in Table 10. We then derive a binary label $\mathrm{bin\_label}$ by thresholding at 0.8: pairs with $\ell > 0.8$ are assigned $\mathrm{bin\_label} = 1$ ("same"), and all others are assigned $\mathrm{bin\_label} = 0$ ("different"). This yields 1,527 negative pairs and 473 positive pairs.

Binary label distribution. Using the above threshold, the binary label counts are: 1,527 pairs with $\mathrm{bin\_label} = 0$ and 473 pairs with $\mathrm{bin\_label} = 1$.

D.3. MLLM Evaluation Criteria

To ensure fair comparison with MLLMs, we employ structured evaluation protocols including DreamBench++. The MLLM* row of Table 2(a) reports results using the original prompts and models from Subjects200K and DreamBench++, reflecting the annotations provided with the released datasets. In particular, DreamBench++ prompts follow a rubric-based protocol designed for identity-consistency scoring. Although the exact filtering prompts

Table 10. **Subjects2k continuous label histogram.** Counts of image pairs falling into each bin of the average human "same" vote fraction ℓ.

Label range	# pairs
$[0.00, 0.09)$	788
$[0.09, 0.18)$	94
$[0.18, 0.27)$	111
$[0.27, 0.36)$	96
$[0.36, 0.45)$	141
$[0.45, 0.55)$	70
$[0.55, 0.64)$	143
$[0.64, 0.73)$	68
$[0.73, 0.82)$	105
$[0.82, 0.91)$	77
$[0.91, 1.00)$	307

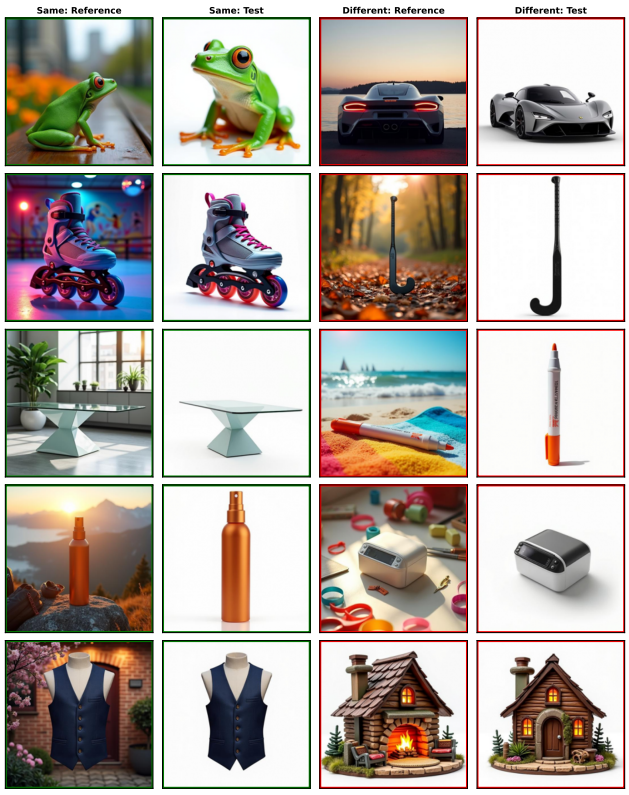

Figure 14. **Human-labeled identity pairs. Left:** Examples annotated as "Same." **Right:** Examples annotated as "Different." Human annotators reliably pick up subtle, fine-grained cues—such as texture, pattern, and small structural differences.

for Subjects200K are not publicly released, the authors describe a rigorous MLLM-based quality control process that explicitly verifies subject consistency. During dataset construction, each generated sample "underwent five independent evaluations by ChatGPT-4o," and "only images that passed all five evaluations were included" in the final dataset [19].

Prompt. You are a visual identity metric. Given two input images, decide if they depict the same instance (e.g., the same animal individual or the same exact object). Focus on stable, instance-specific features and ignore differences due to pose, background, and lighting.

Output format: Return **only** a single JSON object with exactly these fields:

{
`"same_instance": 0 or 1,` // binary decision (1 = same, 0 = different)
`"confidence": float in [0,1],` // confidence in the decision
`"similarity": float in [0,1]` // similarity score for ranking/mAP (higher is more similar)
}

Figure 15. GPT-Generated prompt used for MLLM standardized evaluation.

In addition to the released prompts, we evaluate MLLM-based scoring using a standardized evaluation prompt with additional models, including GPT-5 and Gemini-2.5. This allows us to compare methods under a consistent evaluation setting and extend evaluation to newer models that were not part of the original benchmarks. The prompt instructs the model to act as an identity metric and is applied uniformly across datasets; the full prompt is shown in 15.

E. Results

E.1. Dense Results

We show qualitative results for instance segmentation below, comparing ID-Sim and DINOv3.

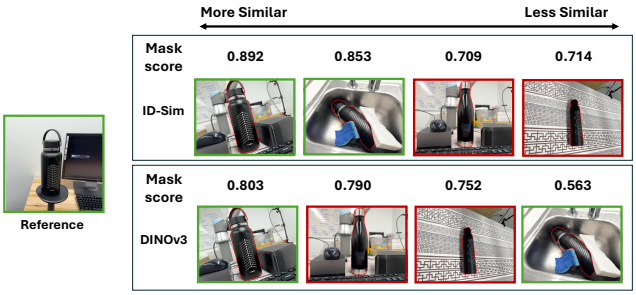

Figure 16. **Qualitative Results for Per-SAM.** We show predicted segmentation masks and corresponding predicted confidence scores, ordered in highest to lowest with respect to a reference object. First, we observe that when combined with PerSAM, both ID-Sim and DINOv3 are able to produce reliable segmentation mask predictions (mask drawn in red around the instance). However, we observe that ID-Sim is significantly better at recognizing instances across distribution shifts and discriminating fine-grained neighbors compared to DINOv3, as shown by the predicted mask scores and the ordering.

In addition to being useful for spatial tasks, ID-Sim's dense features can be integrated with additional conditioning to extend ID-Sim's capabilities in more complex scenarios. This is clearly demonstrated in 17, where we can see that ID-Sim *learns spatially-localized identity features* that remain informative even in multi-entity scenes. This is particularly helpful in resolving ambiguity in multi-entity scenes, which require additional user conditioning to specify which instances the metric should be applied towards. While conditioning is not part of our metric, this shows ID-Sim is naturally compatible with external conditioning signals (e.g., spatial masks or region selection) for specifying user intent.

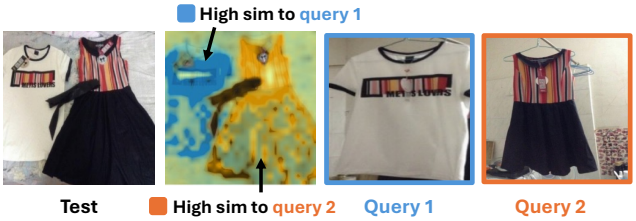

Figure 17. **Dense masks can resolve ambiguities in multi-object Scenes.** Given the test image with 2 shirts (left), ID-Sim features are sensitive to the identity of the query image (right 2 images), evidenced by the patch-level similarity heatmaps (2nd to left).

E.2. Full results

In Figure 19, we report full numerical results across all datasets and baselines. Beyond the metrics shown in the main paper, we include additional evaluation metrics and settings for several benchmarks. For ID-Sim, we also report variance over 10 independent runs, each trained with a different random seed.

F. Analysis

We use 100 held-out object instances from MVImgNet, a multi-view dataset that does not appear in any training or evaluation set. For each object, we generate a dense grid of edited images that vary jointly along identity change and one additional dimension (background, viewpoint, or lighting). This provides controlled perturbations for measuring how similarity scores behave under specific visual changes.

Per-instance regression. For each instance, we fit a linear model to all similarity scores $\{\text{sim}_i\}$:
$\text{sim}_i = \beta_0 + \beta_1 \,\text{factor-change}_i + \beta_2 \,\text{identity-change}_i + \varepsilon_i$.

Sensitivity to each dimension is defined as the negative slope ($-\beta_1$ or $-\beta_2$), which gives the amount of similarity reduction per unit change. Because the regression uses all points in the joint edit grid, it produces stable directional sensitivity estimates while avoiding the noise that arises when using only axis-restricted slices. We also record the regression R^2 value for each instance.

Aggregation across objects. Dataset-level sensitivities are obtained by averaging per-instance slopes across the 100 MVImgNet objects. The same grid construction, regression fitting, and aggregation are applied independently to each evaluated model.

Bootstrap uncertainty. To estimate uncertainty, we perform bootstrap resampling over object identities. In each of 1,000 bootstrap iterations, we sample the 100 instances with replacement, recompute all regression coefficients, and compute the mean sensitivity for that resample. For each model and each dimension, we report:

- the bootstrap mean,
- the bootstrap standard deviation,
- the 95% confidence interval (2.5 to 97.5 percentile).

These intervals capture variability across object identities and provide a reliable measure of uncertainty for the estimated sensitivities.

F.1. Analysis Image Generation

We generate three types of edits (identity, lighting, and background) using dedicated Qwen-Image-Edit pipelines. These images are used only for sensitivity analysis and are fully separate from all training data.

Identity edits. For each anchor image:

- Qwen-Image-Edit operates in inpainting mode over the foreground mask,
- We use seven edit strengths $0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$ to produce a graded sequence of identity change,
- The prompt instructs Qwen to alter internal appearance while preserving overall structure and silhouette,
- Only the foreground region is modified while the background remains unchanged,
- A fixed seed is used for reproducibility.

Lighting edits. Lighting variations are generated with global edits (no masking):

- Eight lighting prompts (shown below)
- Prompts adjust illumination, color temperature, and shading while keeping geometry and texture fixed,
- Qwen-Image-Edit is run with 8 inference steps and a fixed seed.

Background edits. Background replacements are created with mask-guided editing:

- The background is removed using a mask and replaced with a white canvas prior to editing,

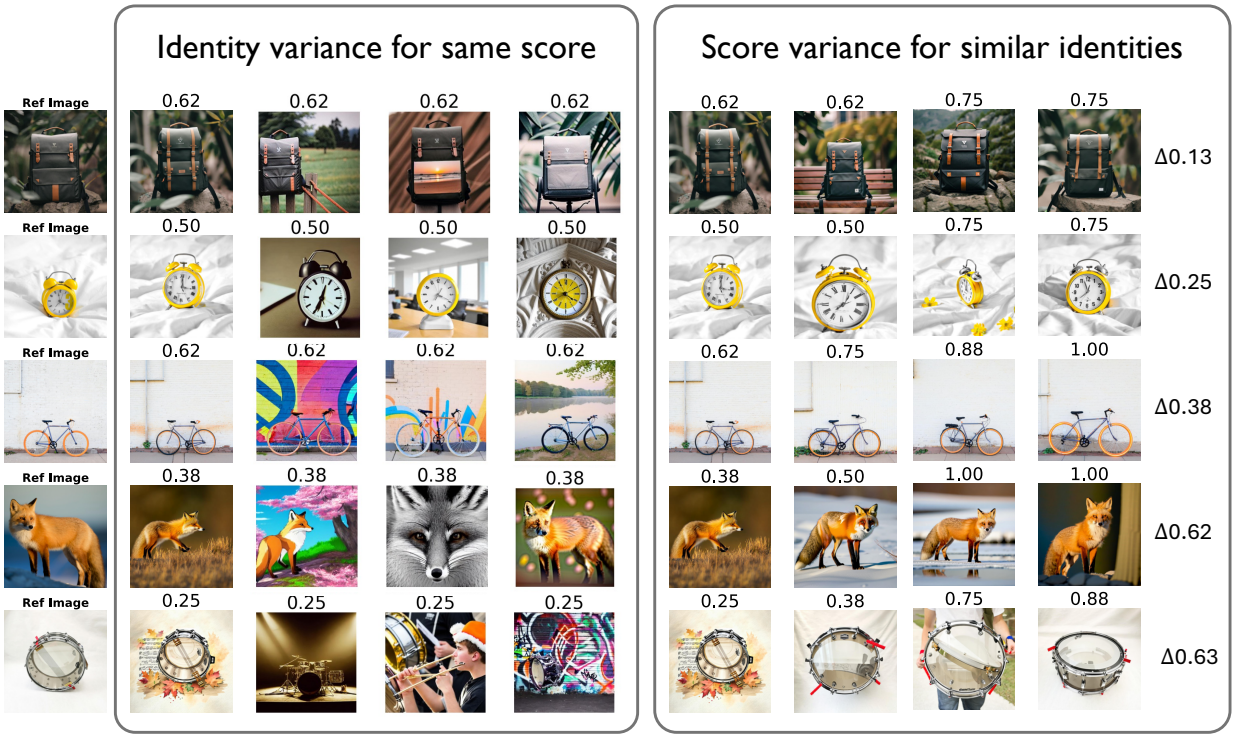

Figure 18. **Limitations of DreamBench++ annotations.** DreamBench++ assigns only two human rubric scores (0–4) per image, which leads to substantial noise in concept-preservation evaluation. As shown above, (i) images with the same DreamBench score can exhibit large variation in identity similarity, and (ii) images with high identity similarity may still receive widely different DreamBench scores. These inconsistencies highlight the need for a cleaner and more discriminative human benchmark, motivating the construction of our Subjects2k dataset.

	PODS			DeepFashion2	AerialCattle	PetFace		CUTE		SS200k		DreamBench	
Model	AUC	AP AUC	nDCG	mAP	mAP	AUC	AP AUC	Easy Acc	Hard Acc	AUC	AP AUC	Spear.	Kend.
Foundation models													
DINOv3	0.929	0.424	0.744	0.519	0.516	0.879	0.884	0.827	0.813	0.642	0.323	0.576	0.437
CLIP	0.862	0.294	0.656	0.408	0.368	0.754	0.776	0.779	0.687	0.594	0.296	0.625	0.478
OpenCLIP	0.887	0.359	0.705	0.488	0.430	0.753	0.772	0.796	0.699	0.584	0.294	0.666	0.516
Perceptual similarity models													
DreamSim	0.897	0.317	0.672	0.529	0.593	0.814	0.824	0.770	0.734	0.603	0.289	**0.716**	**0.561**
LPIPS	0.603	0.067	0.387	0.309	0.442	0.752	0.769	0.651	0.625	0.483	0.235	0.482	0.354
Instance retrieval model													
UNED	0.944	0.671	0.871	0.714	0.468	0.784	0.800	0.815	0.777	0.654	0.356	0.672	0.523
Ours													
ID-Sim	**0.9642** ± 0.0035	**0.7727** ± 0.0106	**0.9161** ± 0.0050	**0.8045** ± 0.0119	**0.6786** ± 0.0123	**0.9002** ± 0.0072	**0.8958** ± 0.0101	**0.8887** ± 0.0077	**0.8559** ± 0.0124	**0.7053** ± 0.0048	**0.4113** ± 0.0060	0.6856 ± 0.0103	0.5305 ± 0.0100

Figure 19. **Full quantitative comparison across all benchmarks.** We report complete numerical results for all datasets and baselines. For ID-Sim, we show mean ± standard deviation over 10 independent training runs. All evaluations use the CLS embedding at inference, consistent with the main paper.

- Eleven background prompts of varying intensity (see below)
- The prompt specifies that only background pixels may change and that the object must remain unchanged in geometry, pose, and fine appearance,
- Qwen adjusts shading to maintain foreground and background consistency,
- Deterministic seeds produce reproducible outputs.

These edit types provide controlled and interpretable variations for quantifying how models respond to identity changes, contextual changes, and illumination changes.

Prompt sets used for analysis. For completeness, we list the exact background and lighting prompts used to generate the edit grids described in this section. These prompts correspond directly to the options indexed in our code and are referenced when constructing the background–identity grid, the lighting–identity grid, and the viewpoint–identity grid.

Background prompts (11).

1. Soft matte off-white plaster wall with subtle imperfections and even diffused daylight.
2. Coastal scene with overcast bright daylight, pale sandy boardwalk, soft gray ocean, and light cloudy sky.
3. Contemporary office interior with white walls, light wood, glass partitions, and soft diffuse daylight.
4. Indoor greenery in white pots with bright filtered daylight from a large window and light walls.
5. Urban street wall with faded or pastel graffiti on light concrete under overcast daylight.
6. Artistic studio with neutral-toned canvases, minimal paint splatter, and soft shadow-free daylight.
7. Bright modern kitchen with white or light-gray surfaces, minimal decor, and soft natural daylight.
8. Minimalist boutique or gallery space with light walls, wood floors, neutral displays, and even ambient lighting.
9. Sunlit desert landscape with pale sand, warm beige rock formations, and soft shadows under clear daylight.
10. Industrial loft with light-exposed brick, large windows with daylight, and pale metal beams.
11. Warm-toned library interior with light wood shelves, muted books, and soft warm ambient lighting.

Lighting prompts (8).

1. Neutral, balanced front lighting with soft shadows and natural highlights (reference lighting).
2. Warm front-directional lighting with soft elongated shadows and a gentle amber color cast.
3. Strong directional side lighting with pronounced contrast between lit and shaded regions.
4. Bright neutral-cool lighting with soft-edged shadows and crisp highlights.
5. Extremely soft front lighting with faint highlights and very low-contrast shadows.
6. Bright front lighting with well-defined shadows and accentuated surface detail.
7. Very low-level front illumination that preserves shape and color with shadow dominance.
8. Intense front lighting with high brightness, strong highlights, and deep detailed shadows.

These prompts define the discrete levels of background variation and illumination used to construct the edit grids in Figures 20 and 22. They are applied consistently across all MVImgNet objects to ensure comparable and fully reproducible sensitivity measurements.

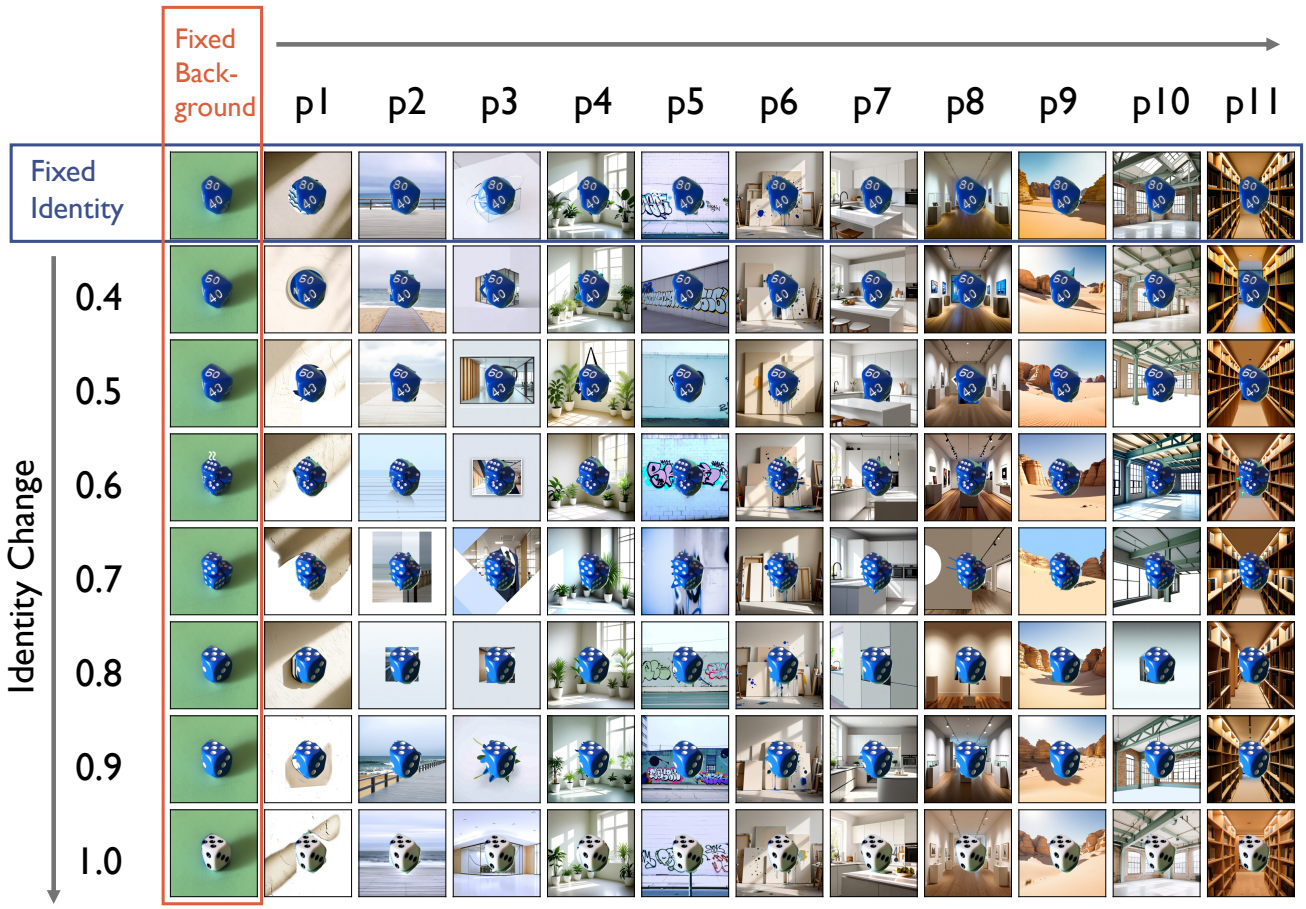

Figure 20. **Background vs. Identity Variation Grid.** Rows vary the foreground identity through Qwen-Edit inpainting at increasing edit strengths, while columns vary the scene background using inpainting prompts. Each cell shows the similarity of the edited image to the original anchor. This grid isolates how models respond jointly to identity changes and background shifts.

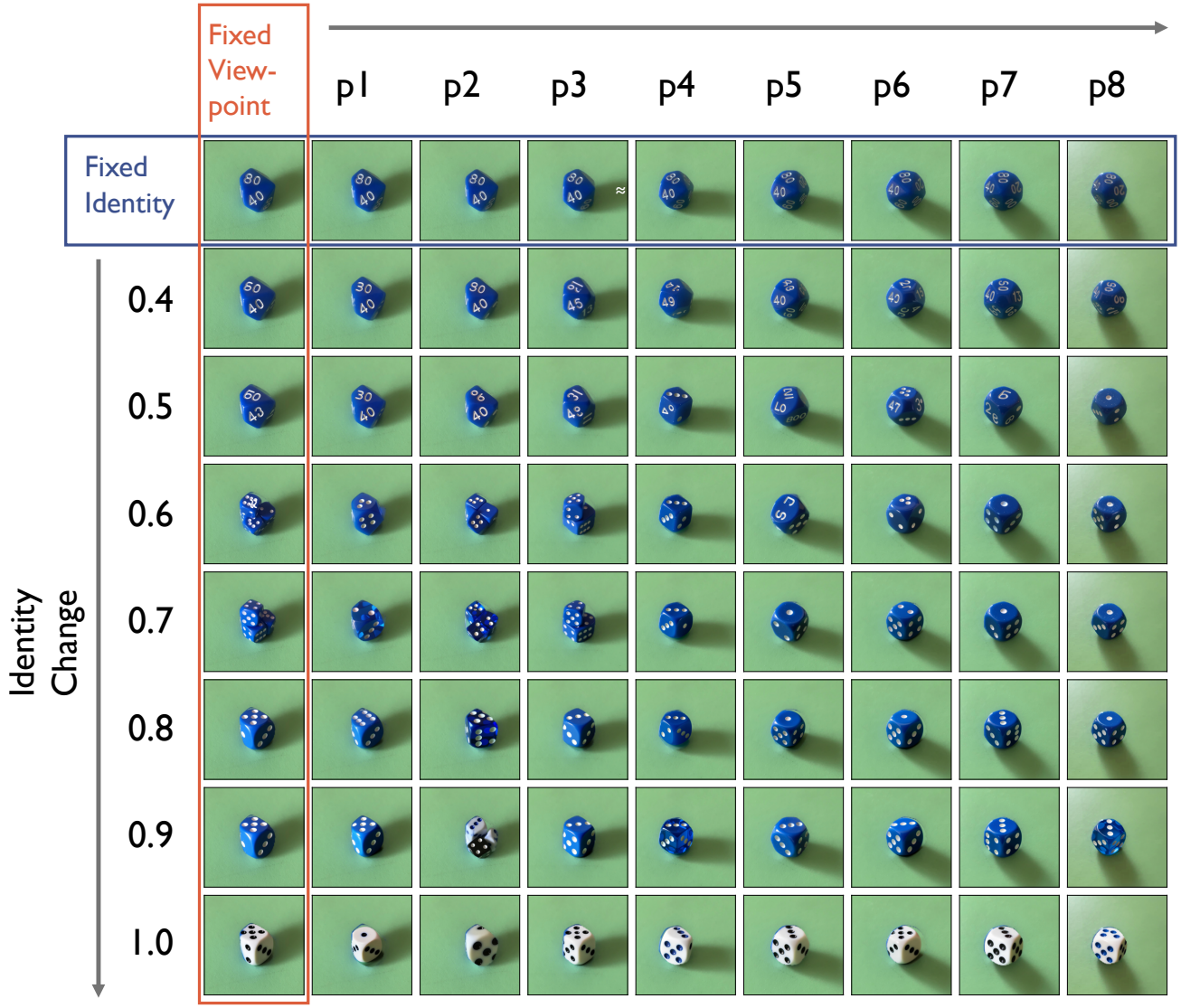

Figure 21. **Viewpoint Variation Grid.** Rows vary identity strength and columns sweep natural viewpoint changes using the multi-view MVImgNet sequence. This grid evaluates how well each model maintains invariance to viewpoint while still detecting identity-altering edits.

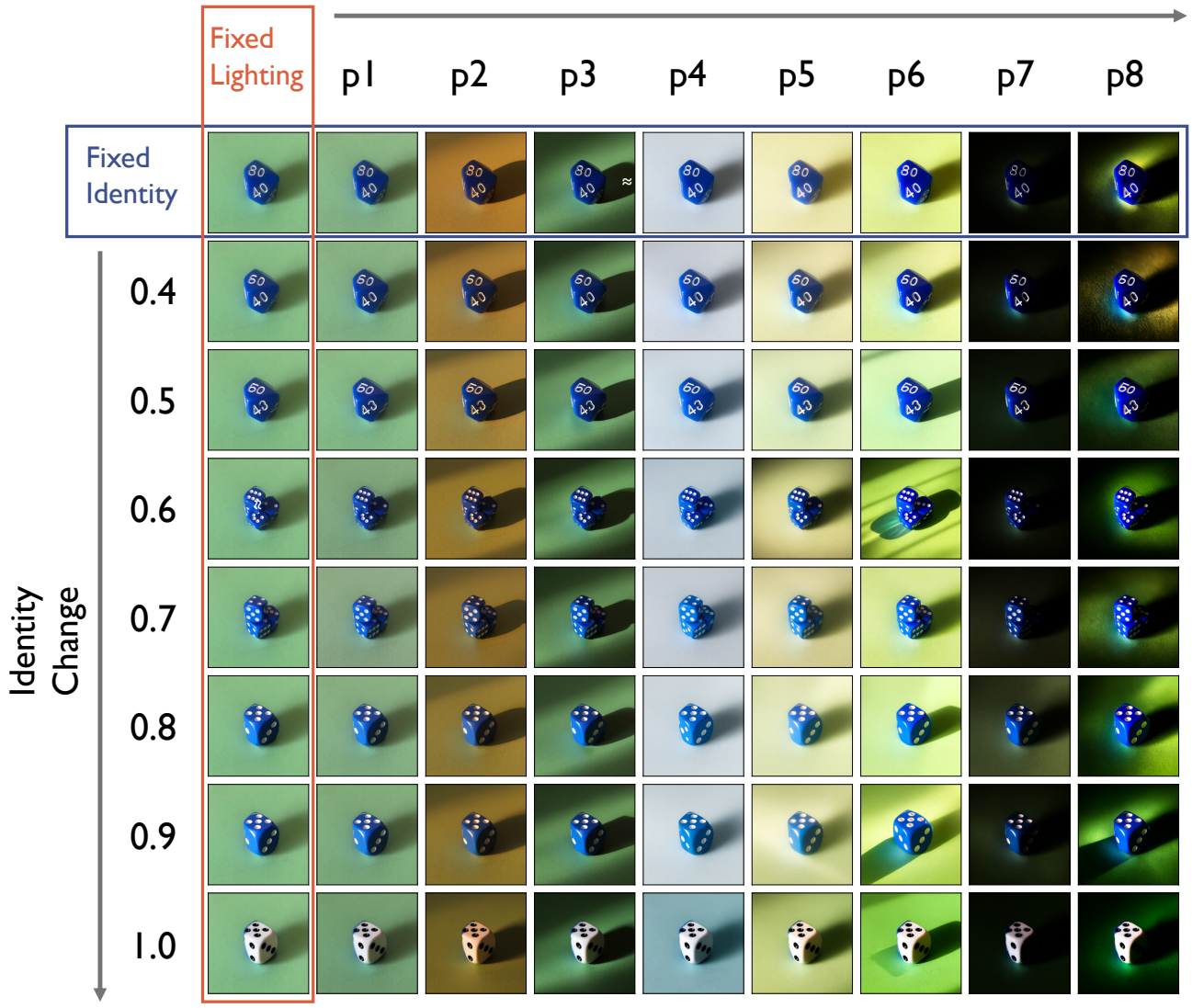

Figure 22. **Lighting Variation Grid.** Rows correspond to increasing levels of identity change, while columns apply eight different lighting edits using Qwen-Edit. This grid tests whether models remain stable under illumination changes while remaining sensitive to small identity perturbations.

References

[1] Lukáš Adam, Vojtěch Čermák, Kostas Papafitsoros, and Lukas Picek. Wildlifereid-10k: Wildlife re-identification dataset with 10k individual animals. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2090–2100. IEEE, 2025. 1

[2] Adobe Inc. Adobe photoshop. 6

[3] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark, 2020. 3

[4] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5337–5345, 2019. 1, 5

[5] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2021. 3

[6] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 4, 10

[7] Giorgos Kordopatis-Zilos, Vladan Stojnić, Anna Manko, Pavel Suma, Nikolaos-Antonios Ypsilantis, Nikos Efthymiadis, Zakaria Laskar, Jiri Matas, Ondrej Chum, and Giorgos Tolias. Ilias: Instance-level image retrieval at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14777–14787, 2025. 1

[8] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 5, 10

[9] Xingchen Liu, Piyush Tayal, Jianyuan Wang, Jesus Zarzar, Tom Monnier, Konstantinos Tertikas, Jiali Duan, Antoine Toisoul, Jason Y. Zhang, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, and David Novotny. Uncommon objects in 3d. In *arXiv*, 2024. 3, 5

[10] Yuang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 9

[11] Adrian Rosebrock. Blur detection with opencv. https://pyimagesearch.com/2015/09/07/blur-detection-with-opencv/, 2015. Accessed: 2021-07-12. 3

[12] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding, 2015. 1

[13] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 1

[14] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. 3

[15] Pengxiang Wu, Siman Wang, Kevin Dela Rosa, and Derek Hu. Forb: a flat object retrieval benchmark for universal image embedding. *Advances in Neural Information Processing Systems*, 36:25448–25460, 2023. 1

[16] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation, 2019. 3

[17] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021. 1

[18] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 1

[19] Xingyi Yang Qiaochu Xue Zhenxiong Tan, Songhua Liu and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024. 9, 10, 11