Supplementary Material: What's in a Latent? Leveraging Diffusion Latent Space for Domain Generalization

A. Is clustering necessary?

With a motive to understand the role of clustering of features from Ψ before feature concatenation, we conduct an empirical analysis comparing GUIDE with and without pseudo-domain clustering. To this end, we directly append the raw features $\mathbf{\Psi}(x)$ to $\Phi(x)$. This results in a moderate gain of $+1.3$ over ERM, whereas clustering improves performance by $+\mathbf{3.3}$ on the PACS dataset. We believe that clustering helps smooth out any noise or sample-specific variations and creates more stable (pseudo) domain representations. Clustering also offers more interpretability to inspect what domain-specific variations are captured in the latent space (Fig. 4).

B. Transformation Function

Transformation ($\mathcal{T}$)	**Acc**
ERM	83.8
Direct Concatenation (No Transformation)	84.3
Cluster-Based Replacement	84.6
Linear Regression	85.7
RBF Kernel Ridge Regression	**87.1**

Table 9. **Effect of $\mathcal{T}$ on Test Accuracy for PACS, using GUIDE-DiT.** We find that the RBF step (Sec 4.1) aids in classification performance on unseen domains.

Effect of the choice of $\mathcal{T}$:

As noted in Sec. 3.3, we apply a transformation function $\mathcal{T} : \mathbf{\Psi} \mapsto \mathbf{\Phi}$ to bring the latent manifold of Ψ closer to Φ and mitigate feature domain drift. To understand the role of $\mathcal{T}$, we explore the following alternatives to it:

- **(a) Direct concatenation**, i.e., appending pseudo-domain representations (from $\mathbf{\Psi}$) to the features (from $\mathbf{\Phi}$) without any transformation. While this introduces domain-specific information, lack of alignment between the two feature spaces led to a minimal improvement of $+0.5\%$ over ERM.
- **(b) Cluster-based replacement**, where pseudo-domains identified in the $\mathbf{\Psi}$ space are used to compute cluster centroids using features from $\mathbf{\Phi}$ space, i.e. cluster samples are averaged in $\mathbf{\Phi}$ space. This provides a slightly better alignment yielding an accuracy gain of $+0.8\%$ over the baseline.
- **(c) Linear regression**, where a linear mapping is learned between the pseudo-domain centroids and the centroids obtained in **(b)**. This helps in bridging differences between $\mathbf{\Psi}$ and $\mathbf{\Phi}$ better, leading to a larger improvement of $+1.4\%$.
- **(d) RBF kernel ridge regression**, where the linear regressor in **(c)** is replaced with an RBF kernel (Sec 4.1). We note that this achieves the highest accuracy gains of $+3.3\%$, highlighting its effectiveness of bridging feature domain drift while incorporating pseudo-domain information into the classifier.

These results underscore the necessity of a well-chosen transformation to fully leverage the pseudo-domain information.

C. Domain Predictability

Dataset	DiT	SD-2.1	MAE	CLIP	DINOv2	RN50
PACS	98.89	**98.95**	98.69	98.29	98.89	97.85
VLCS	**96.08**	92.72	94.03	83.87	81.86	88.48
TerraInc	**99.97**	99.94	99.91	99.83	99.87	99.79
OfficeHome	**89.16**	86.43	82.55	83.41	78.28	77.52
DomainNet	88.55	**89.58**	87.50	87.61	87.24	87.21
Synth-Artists	**100**	99.00	97.00	92.00	90.00	97.00
Synth-Photography	83.33	**87.50**	86.67	73.33	78.33	77.50

Table 10. **Comparison of Domain Predictability Scores Across Datasets.** Diffusion models consistently outperform other models in domain predictability scores, highlighting the effectiveness of encoding domain-specific information in their latent space.

Domain Predictability: To complement NMI, we evaluate domain predictability and predict domain labels from latent feature representations. Specifically, we use a single-layer MLP classifier, trained on an 80-20 train-test split. We report the mean test accuracy over 3 such random splits. While NMI measures alignment and variance across samples belonging to a domain, domain predictability directly assesses a latent representation's ability to learn to classify domain information. We observe in Table. 10 that diffusion models attain the highest domain predictability scores, highlighting their effectiveness in encoding domain-specific information.

D. Label Noise and Domain Inconsistencies

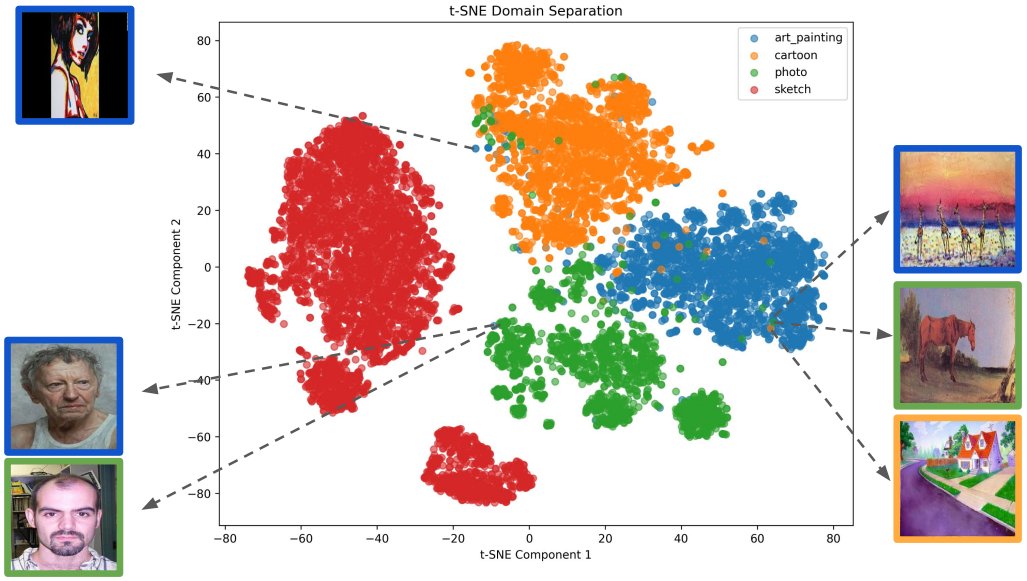

Figure 6. **Examples of inconsistent or confusing domain labels**. Given that most datasets in this study are web-scraped, we expect there to be label noise and domain inconsistencies which may impact the NMI scores. These examples from the PACS dataset and SD-2.1 feature space illustrate cases where domain assignments may be unclear or conflicting. The color of the border on the images denotes the ground truth domain label.

E. Effect of Text-Conditioning in SD-2.1 for Domain Separation

Dataset	Domain NMI		Domain Predictability	
	Empty Prompt	Prompt	Empty Prompt	Prompt
PACS	0.82	0.85	98.95	99.51
OfficeHome	0.22	0.24	86.43	92.91

Table 11. **Domain NMI and predictability scores for empty vs text conditioned prompts for SD-2.1 on PACS and OfficeHome.** For text conditioning we used the prompt: "A photo of an object in the style of {domain}". Similar to the findings of Kim et al. [32], text conditioning appears to activate more relevant features.

F. Effect of Layer and Timestep in Diffusion Models for Domain Separation (DiT vs SD-2.1) on PACS, and VLCS

Following Kim et al. [32], we choose a lower noise level at timestep (t=50), with a motivation to capture rich fine-grained visual information. We use t=50 for both DiT (at block 14) and SD-2.1 (at `up_ft:1`) for both class and domain NMI scores (in Tables 3, and 4), and to obtain the classification accuracies in Table 6. In Fig. 7, we observe that t=50 provides the highest domain NMI score for PACS using DiT. We also note that on VLCS, the `bottleneck` layer outperforms the domain NMI score obtained from `up_ft:1` in Fig. 8, likely due it's focus on coarse-grained features as noted in [32].

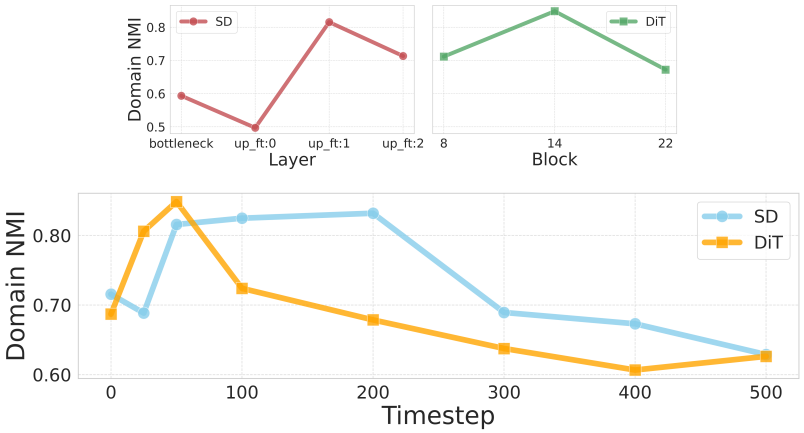

Figure 7. **Domain NMI comparison across layers and timesteps for PACS.** Top: Domain NMI scores for SD-2.1 layers (best: up_ft:1) and DiT blocks (best: block:14). Bottom: Domain NMI scores across various denoising timesteps for SD-2.1 and DiT on PACS.

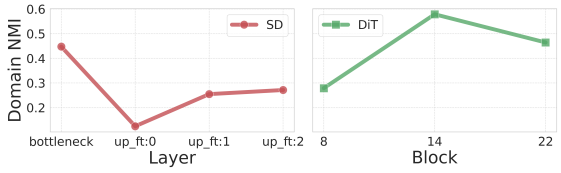

Figure 8. **Domain NMI comparison across layers for VLCS.** The Bottleneck Layer of Stable Diffusion (SD-2.1) which capture more coarse-grained features aids in separating high-level domain shifts in VLCS. However, DiT's superior capability to capture global context via self-attention outperforms the domain NMI scores at `bottleneck` and `up_ft:1`.

G. GUIDE Pseudo-code

Algorithm 1 Training Pseudocode with RBF Kernel Ridge Regression

Input: Training data D_{tr}, transform schedule $T_{\text{transform}}$, K: #clusters
Output: $F_{\text{image}}(.;\boldsymbol{\omega})$, $F_{\text{MLP}}(.;\mathbf{W})$, mapping $\mathcal{T}$

Initialize: Compute feature representations $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, initialize model parameters $\boldsymbol{\omega}_0$, $\mathbf{W}$.
$\{\boldsymbol{\psi}_k\}, \{D_k\} \leftarrow \textsc{Clustering}(\boldsymbol{\Psi}, K)$
for $t = 1$ to T **do**
  if $t \in T_{\text{transform}}$ **then**
    For each k: $\widehat{\boldsymbol{\Phi}}_k = \frac{1}{|D_k|} \sum_{\mathbf{x} \in D_k} \boldsymbol{\Phi}(\mathbf{x})$
    Compute pairwise distances $\|\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\|_2, \forall i \neq j$
    $\gamma \leftarrow 1/(2 \cdot \text{median(pairwise distances)}^2)$ {using median heuristic}
    Fit $\mathcal{T}$ **via RBF Kernel Ridge Regression** using $\{\widehat{\boldsymbol{\psi}}_k\} \mapsto \{\widehat{\boldsymbol{\Phi}}_k\}$ and γ
    $\boldsymbol{\psi}'_{\mathbf{x}} \leftarrow \mathcal{T}(\boldsymbol{\psi}_{\mathbf{x}})$
  end if
  for batch $(\mathbf{x}, \boldsymbol{\psi}_{\mathbf{x}}, y)$ in D_{tr} **do**
    $\boldsymbol{\Phi}(\mathbf{x}) \leftarrow F_{\text{image}}(\mathbf{x}; \boldsymbol{\omega}_t)$
    $\boldsymbol{\psi}'_{\mathbf{x}} \leftarrow \mathcal{T}(\boldsymbol{\psi}_{\mathbf{x}})$
    $\hat{y} \leftarrow F_{\text{MLP}}\big(\textsc{Concat}\big(\boldsymbol{\Phi}(\mathbf{x}), \boldsymbol{\psi}'_{\mathbf{x}}\big); \mathbf{W}_t\big)$
    Update $\boldsymbol{\omega}_{t+1}, \mathbf{W}_{t+1}$ via SGD $\textsc{Step}$ on $\mathcal{L} = \textsc{CrossEntropy}(\hat{y}, y)$
  end for
end for
Return $F_{\text{image}}(.;\boldsymbol{\omega}_T)$, $F_{\text{MLP}}(.;\mathbf{W}_T)$, and $\mathcal{T}$

Inference
Input: Test data D_{test}, transformation function $\mathcal{T}$, and centroids $\{\widehat{\boldsymbol{\psi}}_k\}_{k=1}^K$
Output: Predicted labels $\hat{y}$

for $\mathbf{x} \in D_{\text{test}}$ **do**
  $\boldsymbol{\psi}_{\mathbf{x}} \leftarrow \textsc{NearestCentroid}\big(\boldsymbol{\Psi}, \mathbf{x}\big)$ {Find closest cluster in $\boldsymbol{\Psi}$-space}
  $\boldsymbol{\psi}'_{\mathbf{x}} \leftarrow \mathcal{T}(\boldsymbol{\psi}_{\mathbf{x}})$ {Apply same RBF transform as in training}
  $\boldsymbol{\Phi}(\mathbf{x}) \leftarrow F_{\text{image}}(\mathbf{x}; \boldsymbol{\omega}_T)$
  $\hat{y} \leftarrow F_{\text{MLP}}\big(\textsc{Concat}(\boldsymbol{\Phi}(\mathbf{x}), \boldsymbol{\psi}'_{\mathbf{x}}); \mathbf{W}_T\big)$
end for
Return $\hat{y}$

H. Domain Shift Examples and Domain Separation in Feature Spaces

In this section, we provide:

- **Example images**, i.e. class samples across domains for each dataset.
- **Class vs Domain NMI scores** for each feature extractor (Ψ) studied in this work, on each dataset.
- **Feature space visualizations** for each feature extractor (Ψ) studied in this work, on the PACS, VLCS, OfficeHome, and TerraInognita datasets.

H.1. PACS [35]

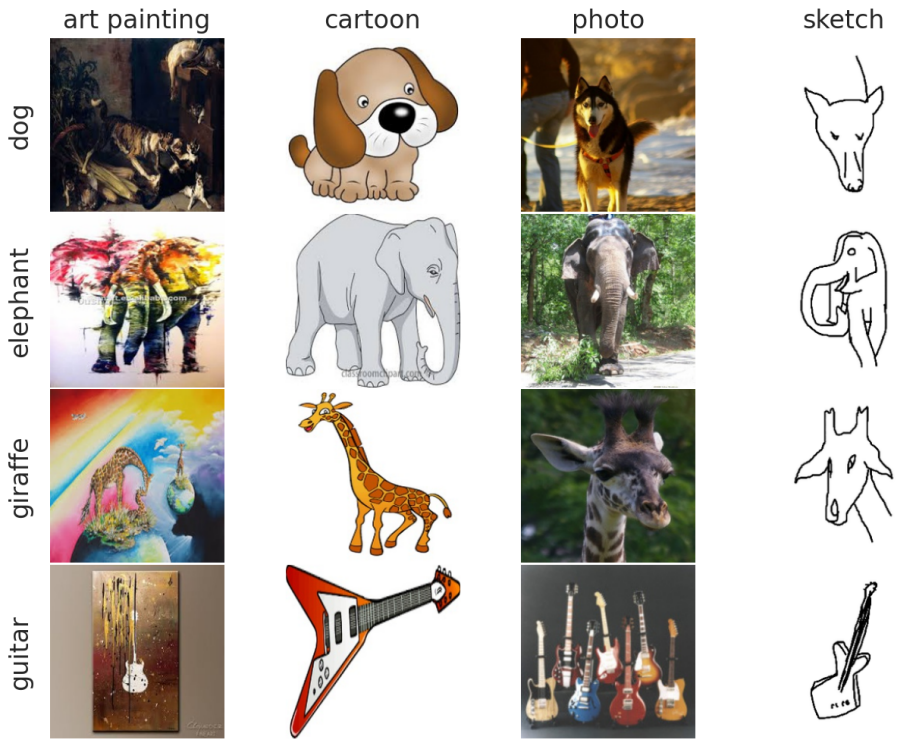

Figure 9. Class examples across domains in the PACS dataset. Each column represents a domain, and each row corresponds to a class.

Domains	Classes
art painting, cartoon, photo, sketch	dog, elephant, giraffe, guitar, horse, house, person

Table 12. 4 domains and 7 classes of the PACS dataset.

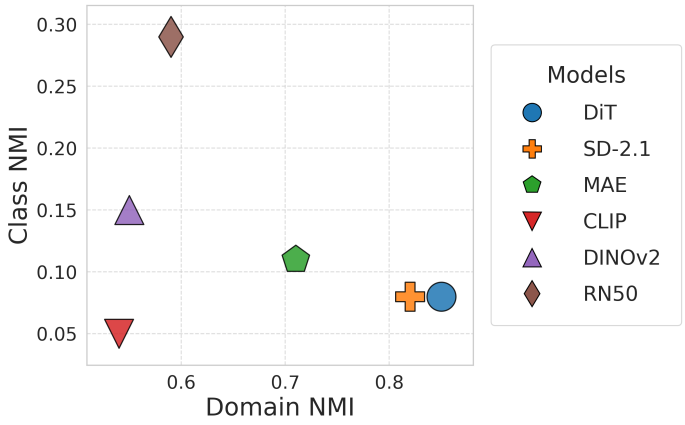

Figure 10. **Class vs Domain NMI scores for PACS.** Note how RN50 has the highest class NMI and diffusion models have low class NMI scores. Diffusion models also has the highest domain NMI scores, thereby capturing domain-specific class invariant structures.

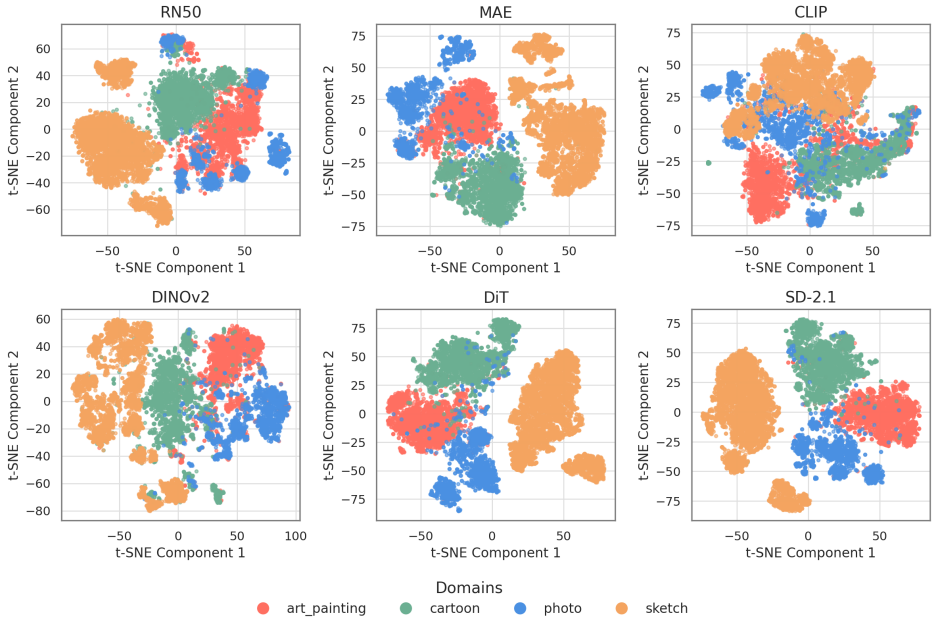

Figure 11. **T-SNE visualization of domain separation for PACS**. Each point represents a sample, colored by its domain. Notice how well separated the domains are when diffusion features are used compared to other models.

H.2. VLCS [17]

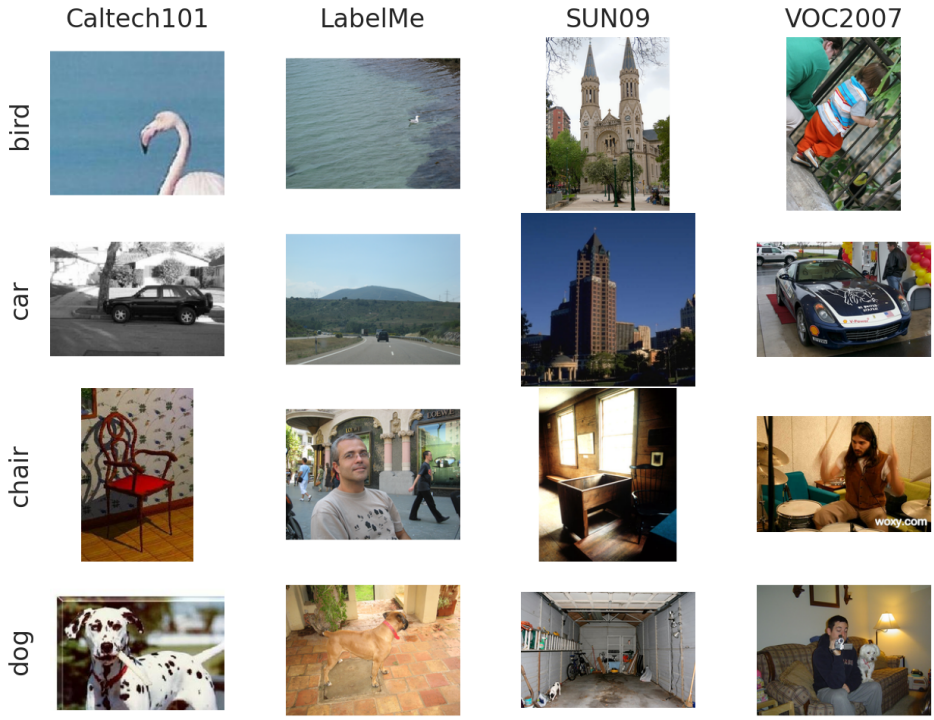

Figure 12. Class examples across domains in the VLCS dataset. Each column represents a domain, and each row corresponds to a class.

Domains	**Classes**
Caltech101, LabelMe, SUN09, VOC2007	bird, car, chair, dog, person

Table 13. 4 domains and 5 classes of the VLCS dataset.

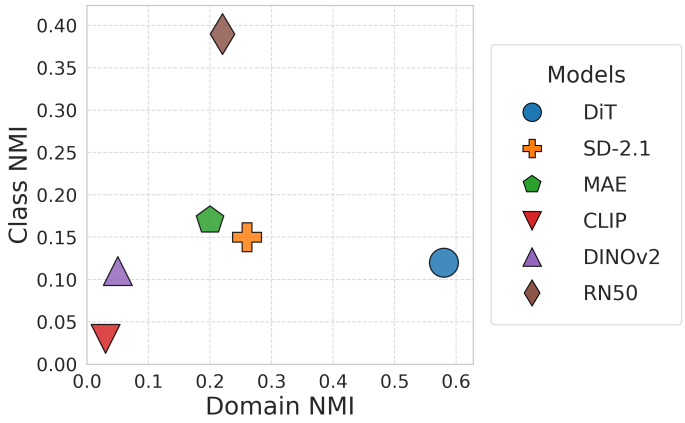

Figure 13. **Class vs Domain NMI scores for VLCS.** Note how RN50 has the highest class NMI score, and diffusion models have low class NMI scores. DiT has a much higher domain NMI score than SD-2.1, resulting from its stronger capability in capturing high-level dataset-specific biases, as discussed in Sec. 4.3.

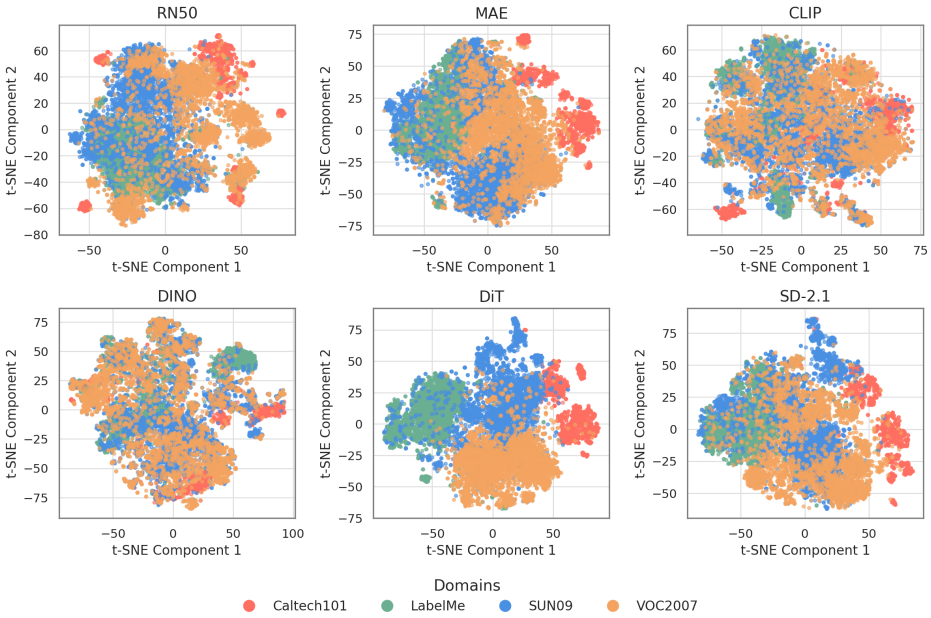

Figure 14. **T-SNE visualization of domain separation for VLCS**. Each point represents a sample, colored by its domain. Note how the DiT feature space best separate the domains.

H.3. OfficeHome [70]

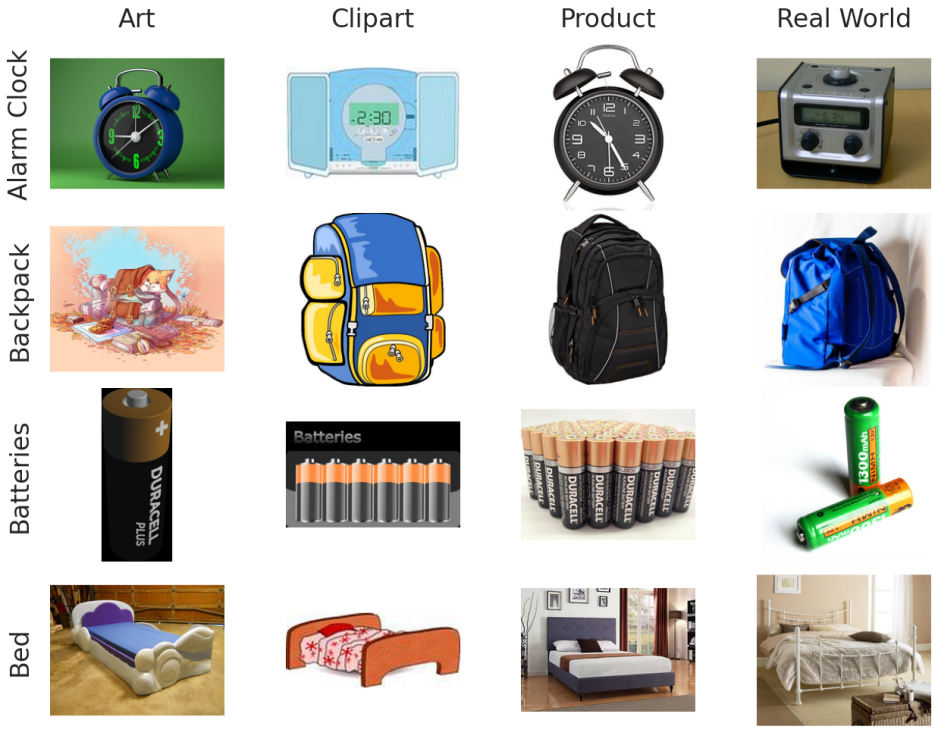

Figure 15. Class examples across domains in the OfficeHome dataset. Each column represents a domain, and each row corresponds to a class.

Domains	**Classes**
Art, Clipart, Product, Real World	Alarm Clock, Backpack, Batteries, Bed, Bike, Bottle, Bucket, Calculator, Calendar, Candles, Chair, Clipboards, Computer, Couch, Curtains, Desk Lamp, Drill, Eraser, Exit Sign, Fan, File Cabinet, Flipflops, Flowers, Folder, Fork, Glasses, Hammer, Helmet, Kettle, Keyboard, Knives, Lamp Shade, Laptop, Marker, Monitor, Mop, Mouse, Mug, Notebook, Oven, Pan, Paper Clip, Pen, Pencil, Post-it Notes, Printer, Push Pin, Radio, Refrigerator, Ruler, Scissors, Screwdriver, Shelf, Sink, Sneakers, Soda, Speaker, Spoon, TV, Table, Telephone, ToothBrush, Toys, Trash Can, Webcam.

Table 14. 4 domains and 65 Classes of the OfficeHome dataset.

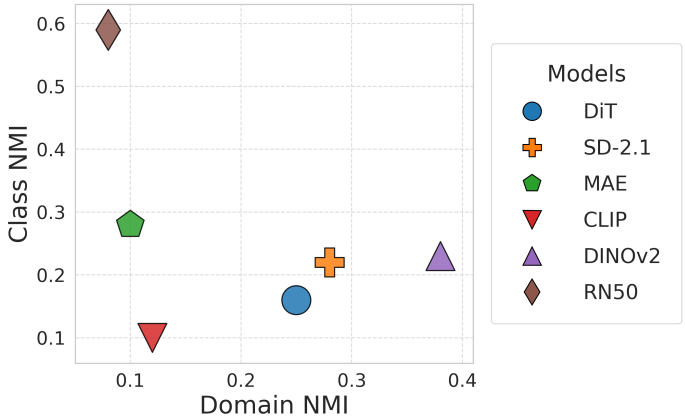

Figure 16. **Class vs Domain NMI scores for OfficeHome.** Note how RN50 has the highest class NMI score and DINOv2 has the highest domain NMI score, resulting form its stronger ability in capturing low-level style shifts, as discussed in Sec. 4.3. DiT and SD-2.1 have moderate domain NMI scores, with DiT having a lower class NMI score.

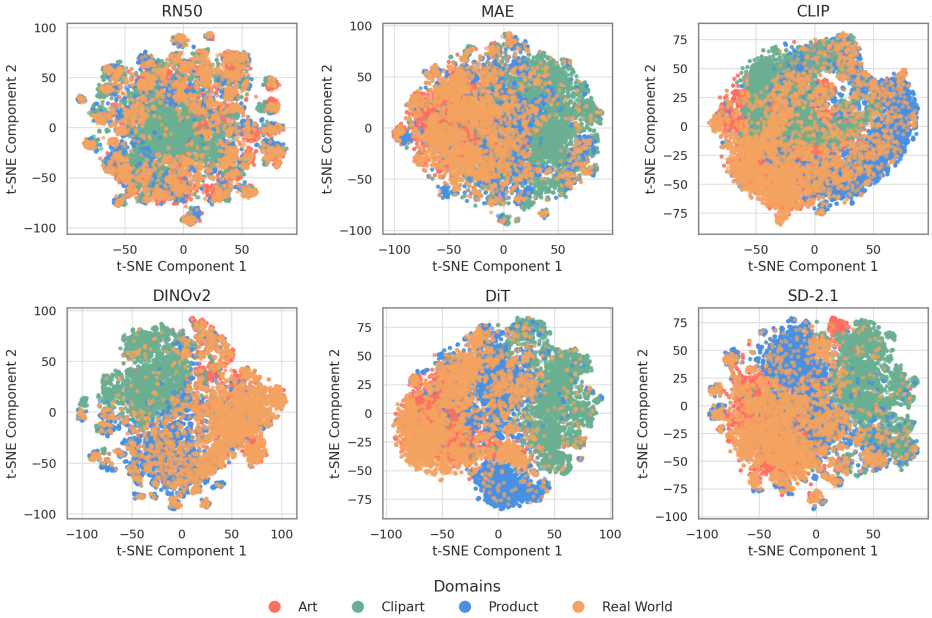

Figure 17. **T-SNE visualization of domain separation for OfficeHome.** Each point represents a sample, colored by its domain. All models struggle to separate the domains in this dataset. The "real" domain has considerable overlap with the other domains.

H.4. TerraIncognita [4]

Figure 18. Class examples across domains in the TerraIncognita dataset. Each column represents a domain, and each row corresponds to a class.

Domains	**Classes**
Location 100, Location 38, Location 43, Location 46	bird, bobcat, cat, coyote, dog, empty, opossum, rabbit, raccoon, squirrel

Table 15. 4 domains and 10 classes of the TerraIncognita dataset.

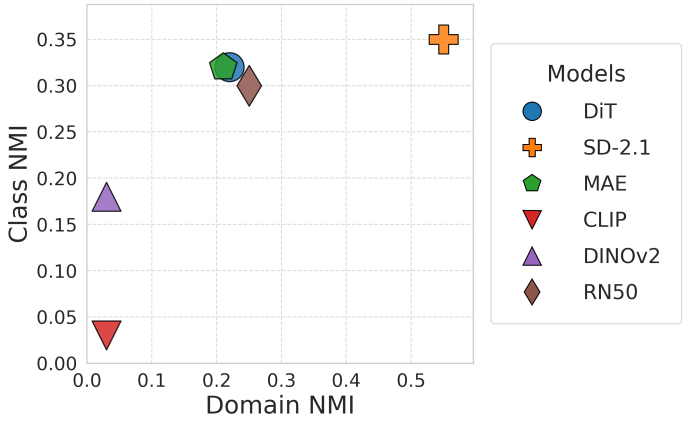

Figure 19. **Class vs Domain NMI scores for TerraIncognita.** Most models have a high class NMI score. SD-2.1 has the highest domain NMI score, resulting from its stronger capability in capturing spatial information, as discussed in Sec. 4.3.

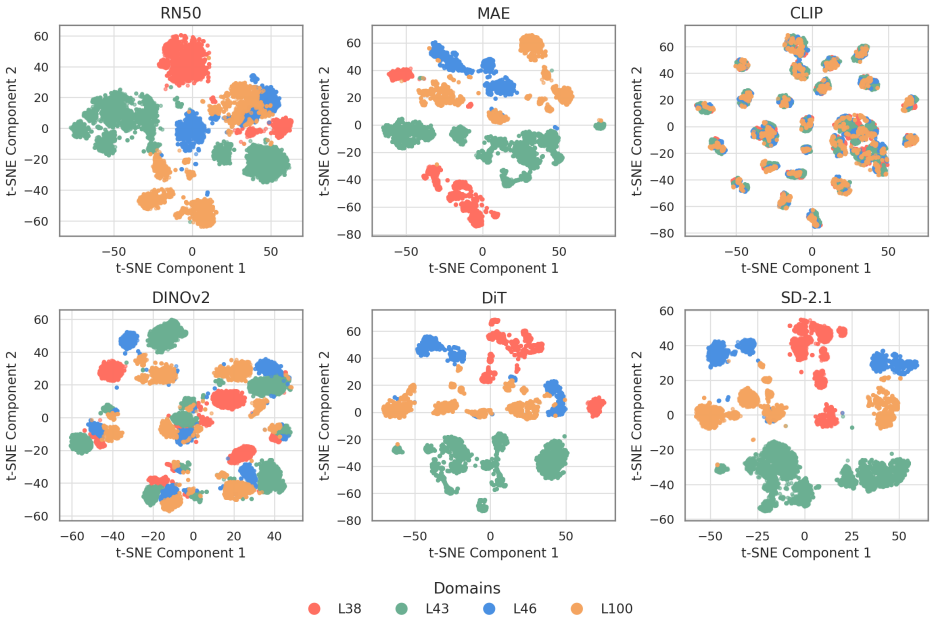

Figure 20. **T-SNE visualization of domain separation for TerraIncognita**. Each point represents a sample, colored by its domain. Note how the SD-2.1 feature space best groups samples from the same domain closer together, and separate from other domains.

H.5. DomainNet [52]

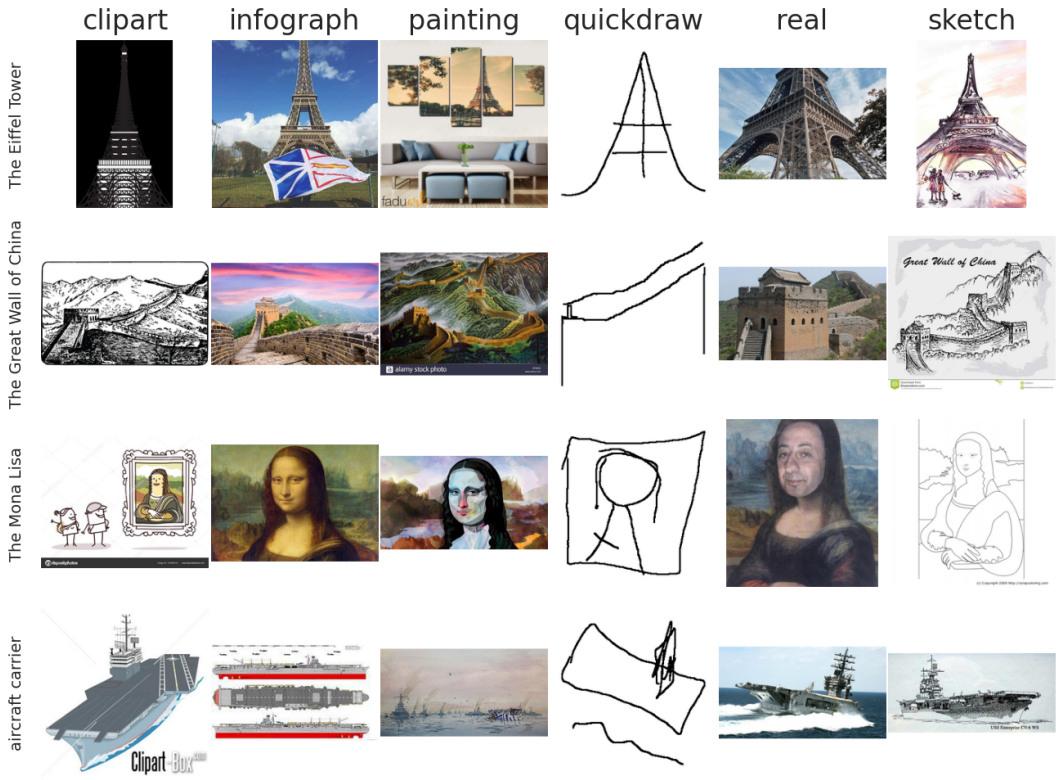

Figure 21. Class examples across domains in the DomainNet dataset. Each column represents a domain, and each row corresponds to a class.

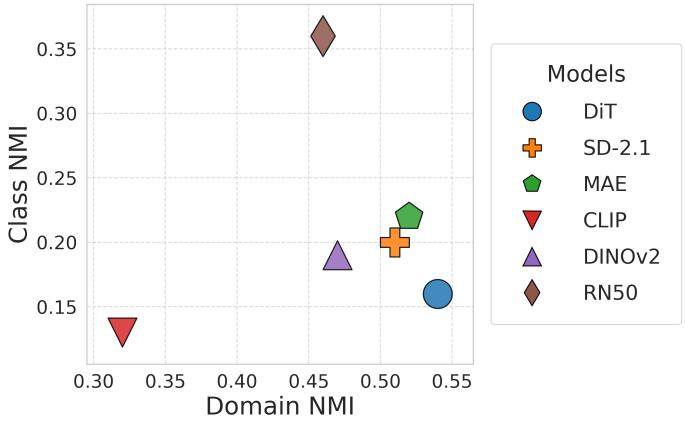

Figure 22. **Class vs Domain NMI scores for DomainNet.** Note how RN50 has the highest class NMI and diffusion models, and MAE have the highest domain NMI scores, with DiT having a lower class NMI score. All models except CLIP exhibit a moderate domain NMI score, likely due to the varied domain shifts inherent in the dataset, as discussed in Sec. 4.3.

Domains	**Classes**
clipart, infograph, painting, quickdraw, real, sketch	The Eiffel Tower, The Great Wall of China, The Mona Lisa, aircraft carrier, airplane, alarm clock, ambulance, angel, animal migration, ant, anvil, apple, arm, asparagus, axe, backpack, banana, bandage, barn, baseball, baseball bat, basket, basketball, bat, bathtub, beach, bear, beard, bed, bee, belt, bench, bicycle, binoculars, bird, birthday cake, blackberry, blueberry, book, boomerang, bottlecap, bowtie, bracelet, brain, bread, bridge, broccoli, broom, bucket, bulldozer, bus, bush, butterfly, cactus, cake, calculator, calendar, camel, camera, camouflage, campfire, candle, cannon, canoe, car, carrot, castle, cat, ceiling fan, cell phone, cello, chair, chandelier, church, circle, clarinet, clock, cloud, coffee cup, compass, computer, cookie, cooler, couch, cow, crab, crayon, crocodile, crown, cruise ship, cup, diamond, dishwasher, diving board, dog, dolphin, donut, door, dragon, dresser, drill, drums, duck, dumbbell, ear, elbow, elephant, envelope, eraser, eye, eyeglasses, face, fan, feather, fence, finger, fire hydrant, fireplace, firetruck, fish, flamingo, flashlight, flip flops, floor lamp, flower, flying saucer, foot, fork, frog, frying pan, garden, garden hose, giraffe, goatee, golf club, grapes, grass, guitar, hamburger, hammer, hand, harp, hat, headphones, hedgehog, helicopter, helmet, hexagon, hockey puck, hockey stick, horse, hospital, hot air balloon, hot dog, hot tub, hourglass, house, house plant, hurricane, ice cream, jacket, jail, kangaroo, key, keyboard, knee, knife, ladder, lantern, laptop, leaf, leg, light bulb, lighter, lighthouse, lightning, line, lion, lipstick, lobster, lollipop, mailbox, map, marker, matches, megaphone, mermaid, microphone, microwave, monkey, moon, mosquito, motorbike, mountain, mouse, moustache, mouth, mug, mushroom, nail, necklace, nose, ocean, octagon, octopus, onion, oven, owl, paint can, paintbrush, palm tree, panda, pants, paper clip, parachute, parrot, passport, peanut, pear, peas, pencil, penguin, piano, pickup truck, picture frame, pig, pillow, pineapple, pizza, pliers, police car, pond, pool, popsicle, postcard, potato, power outlet, purse, rabbit, raccoon, radio, rain, rainbow, rake, remote control, rhinoceros, rifle, river, roller coaster, rollerskates, sailboat, sandwich, saw, saxophone, school bus, scissors, scorpion, screwdriver, sea turtle, see saw, shark, sheep, shoe, shorts, shovel, sink, skateboard, skull, skyscraper, sleeping bag, smiley face, snail, snake, snorkel, snowflake, snowman, soccer ball, sock, speedboat, spider, spoon, spreadsheet, square, squiggle, squirrel, stairs, star, steak, stereo, stethoscope, stitches, stop sign, stove, strawberry, streetlight, string bean, submarine, suitcase, sun, swan, sweater, swing set, sword, syringe, t-shirt, table, teapot, teddy-bear, telephone, television, tennis racquet, tent, tiger, toaster, toe, toilet, tooth, toothbrush, toothpaste, tornado, tractor, traffic light, train, tree, triangle, trombone, truck, trumpet, umbrella, underwear, van, vase, violin, washing machine, watermelon, waterslide, whale, wheel, windmill, wine bottle, wine glass, wristwatch, yoga, zebra, zigzag

Table 16. 6 domains and 325 classes of the DomainNet dataset.

I. Synth-Photography and Synth-Artists Custom Datasets

Figure 23. Synth-Photography examples generated using Stable Diffusion XL [53], each column is a photography effect which forms the domain.

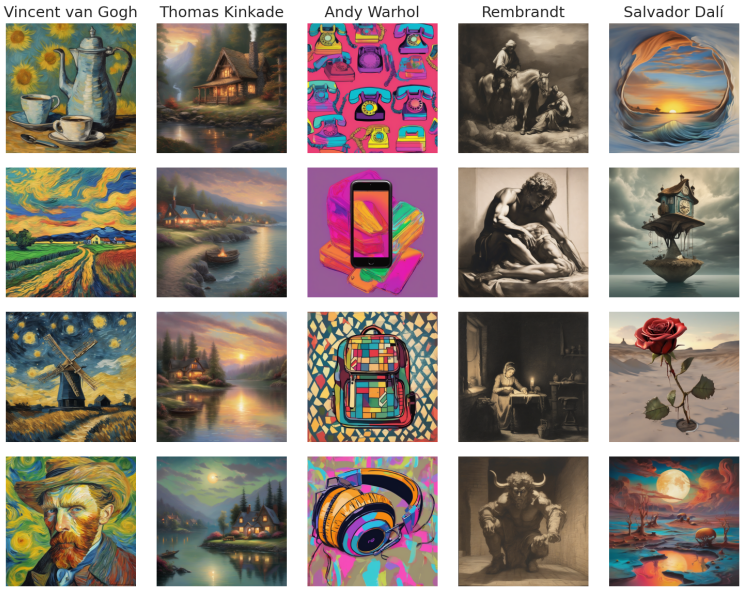

Figure 24. Synth-Artists examples generated using Stable Diffusion XL [53], each column is an artistic style which forms the domain.

We generate the Synth-Photography and Synth-Artists datasets in Sec. 4.5 using Stable Diffusion XL [53]. For Synth-photography (Fig. 23) we use the prompt *"Generate an image in the style of {effect} photography"*; where effect can be Macro, Tilt-Shift, Bokeh, Symmetry, and Zoom Blur. Similarly, for Synth-Artists (Fig. 24) we use the prompt *"Generate an image in the style of {artist}"*; where artist can be Vincent Van Gogh, Thomas Kinkade, Andy Warhol, Rembrandt, and Salvador Dali.