DiverGen: Improving Instance Segmentation by Learning Wider Data Distribution with More Diverse Generative Data

Supplementary Material

A. Implementation Details

A.1. Data Distribution Analysis

We use the image encoder of CLIP [7] ViT-L/14 to extract image embeddings. For objects in the LVIS [3] dataset, we extract embeddings from the object regions instead of the whole images. First, we blur the regions outside the object masks using the normalized box filter, with the kernel size of (10, 10). Then, to prevent objects from being too small, we pad around the object boxes to ensure the minimum width of the padded boxes is 80 pixels, and crop the images according to the padded boxes. Finally, the cropped images are fed into the CLIP image encoder to extract embeddings. For generative images, the whole images are fed into the CLIP image encoder to extract embeddings. At last, we use UMAP [5] to reduce dimensions for visualization. τ is set to 0.9 in the energy function.

To investigate the potential impact of noise in the rare classes to TVG metrics, we conduct additional experiments to demonstrate the validity of TVG. We randomly take five different models each for the LVIS and LVIS + Gen data sources, compute the mean (μ) and standard deviation (σ) of their TVG, and calculate the 3 sigma range ($\mu + 3\sigma$ and $\mu - 3\sigma$), which we think represents the maximum fluctuation that potential noise could induce. As shown in Table 1, we find that: 1) The TVGs of LVIS all exceed the 3 sigma upper bound of LVIS + Gen, while the TVGs of LVIS + Gen are all below the 3 sigma lower bound of LVIS, and there is no overlap between the 3 sigma ranges of LVIS and LVIS + Gen; 2) For both LVIS + Gen and LVIS, there is no overlap between the 3 sigma ranges of different groups, e.g. frequent and common, common and rare. These two findings suggest that even in the presence of potential noise, the results can not be attributed to those fluctuations. Therefore, we think our proposed TVG metrics are reasonable and can support the conclusions.

	TVG_f^{box}	TVG_f^{mask}	TVG_c^{box}	TVG_c^{mask}	TVG_r^{box}	TVG_r^{mask}
μ	9.98	8.60	16.59	13.36	30.23	24.22
σ	0.24	0.18	0.56	0.44	1.12	1.18
$\mu + 3\sigma$	10.70	9.15	18.26	14.69	33.58	27.77
$\mu - 3\sigma$	9.25	8.06	14.91	12.04	26.88	20.68
LVIS	13.16	10.71	21.80	16.80	39.59	31.68

(a) LVIS + Gen

	TVG_f^{box}	TVG_f^{mask}	TVG_c^{box}	TVG_c^{mask}	TVG_r^{box}	TVG_r^{mask}
μ	13.95	11.40	22.53	17.16	43.46	35.10
σ	0.41	0.35	0.43	0.33	1.98	1.75
$\mu + 3\sigma$	15.17	12.45	23.81	18.14	49.39	40.37
$\mu - 3\sigma$	12.73	10.34	21.25	16.17	37.53	29.84
LVIS + Gen	9.64	8.38	15.64	12.69	29.39	22.49

(b) LVIS

Table 1. **Statistics of train-val gap on different data sources.**

A.2. Category Diversity

We compute the path similarity of WordNet [2] synsets between 1,000 categories in ImageNet-1K [9] and 1,203 categories in LVIS [3]. For each of the 1,000 categories in ImageNet-1K, if the highest similarity for that category is below 0.4, we consider the category to be non-existent in LVIS and designate it as an extra category. Based on this method, 566 categories can serve as extra categories. The names of these 566 categories are presented in Table 4.

A.3. Prompt Diversity

Limited by the inference cost of ChatGPT, we use the manually designed prompts as the base and only use ChatGPT to enhance the prompt diversity for a subset of categories. For manually designed prompts, the template of prompts is "a photo of a single {*category_name*}, {*category_def*}, in a white background". category_name and category_def are from LVIS [3] category information. For ChatGPT designed prompts, we select a subset of categories and use ChatGPT to enhance prompt diversity for these categories. The names of the 144 categories in this subset are shown in Table 5. We use GPT-3.5-turbo and have three requirements for the ChatGPT: 1) each prompt should be as different as possible; 2) each prompt should ensure that there is only one object in the image; 3) prompts should describe different attributes of the category. Therefore, the input prompts to ChatGPT contain these three requirements. Examples of input prompts and the corresponding responses from ChatGPT are illustrated in Figure 3. To conserve output token length, there is no strict requirement for ChatGPT designed prompts to end with "in a white background", and this constraint will be added when generating images.

A.4. Generative Model Diversity

We select two commonly used generative models, Stable Diffusion [8] and DeepFloyd-IF [10]. For Stable Diffusion, we use Stable Diffusion V1.5, with 50 inference steps and a guidance scale of 7.5. All other parameters are set to their defaults. For DeepFloyd-IF, we use the output images from stage II, with stage I using the weight IF-I-XL-v1.0 and stage II using IF-II-L-v1.0. All parameters are set to their defaults.

A.5. Instance Annotation

We employ SAM [4] ViT-H as the annotation model. We explore two annotation strategies, namely SAM-foreground and SAM-background. SAM-foreground uses points sampled from foreground objects as input prompts. Specifically, we first obtain the approximate region of the foreground object based on the cross-attention map of the generative model using a threshold. Then, we use k-means++ [1] clustering to transform dense points within the foreground region into cluster centers. Next, we randomly select some points from the cluster centers as inputs to SAM. We use various metrics to evaluate the quality of the output mask and select the mask with the highest score as the final mask. However, although SAM-foreground is intuitive, it also has some limitations. Firstly, cross-attention maps of different categories require different thresholds to obtain foreground regions, making it cumbersome to choose the optimal threshold for each category. Secondly, the number of points required for SAM to output mask varies for different foreground objects. Complex object needs more points than simple object, making it challenging to control the number of points. Additionally, the position of points significantly influences the quality of SAM's output mask. If the position of points is not appropriate, this strategy is prone to generating incomplete masks.

Therefore, we discard SAM-foreground and propose a simpler and more effective annotation strategy, SAM-background. Due to our leveraging of the controllability of the generative model in instance generation, the generative images have two characteristics: 1) each image predominantly contains only one foreground object; 2) the background of the images is relatively simple. SAM-background directly uses the four corner points of the image as input prompts for SAM to obtain the background mask, then inverts the background mask as the mask of the foreground object. The illustrations of point selection for SAM-foreground and SAM-background are shown in Figure 1. By using SAM-background for annotation, more refined masks can be obtained. Examples of annotations from SAM-foreground and SAM-background are shown in Figure 2.

SAM-foreground SAM-background

Figure 1. **Illustrations of point selection for SAM-foreground and SAM-background.**

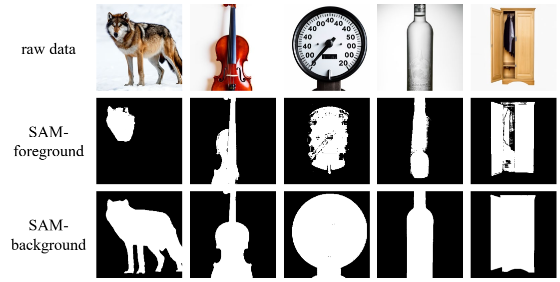


Figure 2. **Examples of annotations from SAM-foreground and SAM-background.** By using SAM-background for annotation, more refined masks can be obtained.

To further validate the effectiveness of SAM-background, we manually annotate masks for some images as ground truth (gt). We apply both strategies to annotate these images and calculate the mIoU between the resulting masks and the ground truth. The results in Table 2 indicate that SAM-background achieves better annotation quality.

Strategy	mIoU
SAM-foreground	0.8163
SAM-background	**0.9418**

Table 2. **Results of SAM-foreground and SAM-background.** SAM-background achieves better annotation quality.

A.6. Instance Filtration

We use the image encoder of CLIP [7] ViT-L/14 to extract image embeddings. The embedding extraction process is consistent with Sec A.1. Then we calculate the cosine similarity between embeddings of objects in LVIS training set and embeddings of generative images. For each generative image, the final CLIP inter-similarity is the average similarity with all objects of the same category in the training set. Through experiments, we find that when the filtering threshold is 0.6, the model achieves the best performance and strikes a balance between data diversity and quality, so we set the threshold to 0.6.

Furthermore, we also explore other filtration strategies. From our experiments, using pure image-trained models like DINOv2 [6] as image encoder or combining CLIP score and CLIP inter-similarity is not as good as using just CLIP inter-similarity alone, as shown in Table 3. Therefore, we ultimately opt to only use CLIP inter-similarity.

Strategy	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
DINOv2	48.02	42.39	40.31	35.27
CLIP score + CLIP inter-similarity	49.82	44.30	45.26	40.92
CLIP inter-similarity	**50.07**	**44.44**	**45.53**	**41.16**

Table 3. **Results of different filtration strategies.**

A.7. Instance Augmentation

In instance augmentation, we use the instance paste strategy proposed by Zhao et al. [11] to increase model learning efficiency on generative data. Each image contains up to 20 pasted instances at most.

The parameters not specified in the paper are consistent with X-Paste [11].

B. Visualization

B.1. Prompt Diversity

We find that images generated from ChatGPT designed prompts have diverse textures, styles, patterns, etc., greatly enhancing data diversity. The ChatGPT designed prompts and the corresponding generative images are shown in Figure 4. Compared to manually designed prompts, the diversity of images generated from ChatGPT designed prompts can be significantly improved. A visual comparison between generative images from manually designed prompts and ChatGPT designed prompts is shown in Figure 5.

B.2. Generative Model Diversity

The images generated by Stable Diffusion and DeepFloyd-IF are different, even within the same category, significantly enhancing the data diversity. Both Stable Diffusion and DeepFloyd-IF are capable of producing images belonging to the target categories. However, the images generated by DeepFloyd-IF appear more photorealistic and consistent with the prompt texts. This indicates DeepFloyd-IF's superiority in image generation quality and controllability through text prompts. Examples from Stable Diffusion and DeepFloyd-IF are shown in Figure 6 and Figure 7, respectively.

B.3. Instance Annotation

In terms of annotation quality, masks generated by max CLIP [11] tend to be incomplete, while our proposed SAM-bg is able to produce more refined and complete masks when processing images of multiple categories. As shown in Figure 8, our proposed annotation strategy can output more precise and refined masks compared to max CLIP.

B.4. Instance Augmentation

The use of instance augmentation strategies helps alleviate the limitation in relatively simple scenes of generative data and improves the efficiency of model learning on the generative data. Examples of augmented data are shown in Figure 9.

References

[1] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proc. Annual ACM-SIAM Symposium on Discrete algorithms*, pages 1027–1035, 2007. 2

[2] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010. 1

[3] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5356–5364, 2019. 1

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander Berg, Wan-Yen Lo, et al. Segment anything. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 4015–4026, 2023. 2

[5] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv: Comp. Res. Repository*, 2018. 1

[6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Research*, 2023. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Mach. Learn.*, pages 8748–8763. PMLR, 2021. 1, 2

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10684–10695, 2022. 1

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115: 211–252, 2015. 1

[10] Alex Shonenkov, Misha Konstantinov, Daria Bakshandaeva, Christoph Schuhmann, Ksenia Ivanova, and Nadiia Klokova. Deepfloyd-if, 2023. 1

[11] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, Weiming Zhang, and Nenghai Yu. X-paste: Revisiting scalable copy-paste for instance segmentation using CLIP and stablediffusion. *Proc. Int. Conf. Mach. Learn.*, 2023. 3

tench	great_white_shark	tiger_shark	electric_ray
stingray	brambling	goldfinch	house_finch
junco	indigo_bunting	American_robin	bulbul
jay	magpie	chickadee	American_dipper
kite_(bird_of_prey)	fire_salamander	smooth_newt	newt
spotted_salamander	axolotl	American_bullfrog	loggerhead_sea_turtle
leatherback_sea_turtle	banded_gecko	green_iguana	Carolina_anole
desert_grassland_whiptail_lizard	agama	frilled-necked_lizard	alligator_lizard
Gila_monster	European_green_lizard	chameleon	Komodo_dragon
Nile_crocodile	triceratops	worm_snake	ring-necked_snake
eastern_hog-nosed_snake	smooth_green_snake	kingsnake	garter_snake
water_snake	vine_snake	night_snake	boa_constrictor
African_rock_python	Indian_cobra	green_mamba	Saharan_horned_viper
eastern_diamondback_rattlesnake	sidewinder_rattlesnake	trilobite	harvestman
scorpion	tick	centipede	black_grouse
ptarmigan	ruffed_grouse	prairie_grouse	peafowl
quail	partridge	sulphur-crested_cockatoo	lorikeet
coucal	bee_eater	hornbill	jacamar
toucan	red-breasted_merganser	black_swan	tusker
echidna	platypus	wallaby	wombat
jellyfish	sea_anemone	brain_coral	flatworm
nematode	conch	snail	slug
sea_slug	chiton	chambered_nautilus	American_lobster
crayfish	hermit_crab	isopod	white_stork
black_stork	spoonbill	great_egret	crane_bird
limpkin	common_gallinule	American_coot	bustard
ruddy_turnstone	dunlin	common_redshank	dowitcher
oystercatcher	albatross	grey_whale	dugong
sea_lion	Chihuahua	Japanese_Chin	Maltese
Pekingese	Shih_Tzu	King_Charles_Spaniel	Papillon
toy_terrier	Rhodesian_Ridgeback	Afghan_Hound	Basset_Hound
Beagle	Bloodhound	Bluetick_Coonhound	Black_and_Tan_Coonhound
Treeing_Walker_Coonhound	English_foxhound	Redbone_Coonhound	borzoi
Irish_Wolfhound	Italian_Greyhound	Whippet	Ibizan_Hound
Norwegian_Elkhound	Otterhound	Saluki	Scottish_Deerhound
Weimaraner	Staffordshire_Bull_Terrier	American_Staffordshire_Terrier	Bedlington_Terrier
Border_Terrier	Kerry_Blue_Terrier	Irish_Terrier	Norfolk_Terrier
Norwich_Terrier	Yorkshire_Terrier	Wire_Fox_Terrier	Lakeland_Terrier
Sealyham_Terrier	Airedale_Terrier	Cairn_Terrier	Australian_Terrier
Dandie_Dinmont_Terrier	Boston_Terrier	Miniature_Schnauzer	Giant_Schnauzer
Standard_Schnauzer	Scottish_Terrier	Tibetan_Terrier	Australian_Silky_Terrier
Soft-coated_Wheaten_Terrier	West_Highland_White_Terrier	Lhasa_Apso	Flat-Coated_Retriever
Curly-coated_Retriever	Golden_Retriever	Labrador_Retriever	Chesapeake_Bay_Retriever
German_Shorthaired_Pointer	Vizsla	English_Setter	Irish_Setter
Gordon_Setter	Brittany_dog	Clumber_Spaniel	English_Springer_Spaniel
Welsh_Springer_Spaniel	Cocker_Spaniel	Sussex_Spaniel	Irish_Water_Spaniel
Kuvasz	Schipperke	Groenendael_dog	Malinois
Dobermann	Miniature_Pinscher	Greater_Swiss_Mountain_Dog	Bernese_Mountain_Dog
Appenzeller_Sennenhund	Entlebucher_Sennenhund	Boxer	Bullmastiff
Tibetan_Mastiff	Great_Dane	St._Bernard	husky
Alaskan_Malamute	Siberian_Husky	Affenpinscher	Samoyed
Pomeranian	Chow_Chow	Keeshond	brussels_griffon
Pembroke_Welsh_Corgi	Cardigan_Welsh_Corgi	Toy_Poodle	Miniature_Poodle
Standard_Poodle	dingo	dhole	African_wild_dog
hyena	red_fox	kit_fox	Arctic_fox
grey_fox	tabby_cat	tiger_cat	Persian_cat
Siamese_cat	Egyptian_Mau	lynx	leopard
snow_leopard	jaguar	cheetah	mongoose

meerkat	dung_beetle	rhinoceros_beetle	fly
bee	ant	grasshopper	cricket_insect
stick_insect	praying_mantis	cicada	leafhopper
lacewing	damselfly	red_admiral_butterfly	monarch_butterfly
small_white_butterfly	sea_urchin	sea_cucumber	hare
fox_squirrel	guinea_pig	wild_boar	warthog
ox	water_buffalo	bison	bighorn_sheep
Alpine_ibex	hartebeest	impala_(antelope)	llama
weasel	mink	black-footed_ferret	otter
skunk	badger	armadillo	three-toed_sloth
orangutan	chimpanzee	gibbon	siamang
guenon	patas_monkey	macaque	langur
black-and-white_colobus	proboscis_monkey	marmoset	white-headed_capuchin
howler_monkey	titi_monkey	Geoffroy's_spider_monkey	common_squirrel_monkey
ring-tailed_lemur	indri	red_panda	snoek_fish
eel	rock_beauty_fish	clownfish	sturgeon
gar_fish	lionfish	academic_gown	accordion
aircraft_carrier	altar	apiary	assault_rifle
bakery	balance_beam	baluster_or_handrail	barbershop
barn	barometer	bassinet	bassoon
lighthouse	bell_tower	baby_bib	boathouse
bookstore	breakwater	breastplate	butcher_shop
carousel	tool_kit	automated_teller_machine	cassette_player
castle	catamaran	cello	chain
chain-link_fence	chainsaw	chiffonier	Christmas_stocking
church	movie_theater	cliff_dwelling	cloak
clogs	spiral_or_coil	candy_store	cradle
construction_crane	croquet_ball	cuirass	dam
desktop_computer	disc_brake	dock	dome
drilling_rig	electric_locomotive	entertainment_center	face_powder
fire_screen	flute	fountain	French_horn
gas_pump	golf_ball	gong	greenhouse
radiator_grille	grocery_store	guillotine	hair_spray
half-track	hand-held_computer	hard_disk_drive	harmonica
harp	combine_harvester	holster	home_theater
honeycomb	hook	gymnastic_horizontal_bar	jigsaw_puzzle
knot	lens_cap	library	lifeboat
lighter	lipstick	lotion	loupe_magnifying_glass
sawmill	messenger_bag	maraca	marimba
mask	matchstick	maypole	maze
megalith	military_uniform	missile	mobile_home
modem	monastery	monitor	moped
mortar_and_pestle	mosque	mosquito_net	tent
mousetrap	moving_van	muzzle	metal_nail
neck_brace	notebook_computer	obelisk	oboe
ocarina	odometer	oil_filter	pipe_organ
oscilloscope	oxygen_mask	palace	pan_flute
parallel_bars	patio	pedestal	photocopier
plectrum	Pickelhaube	picket_fence	pier
pirate_ship	block_plane	planetarium	plastic_bag
plate_rack	plunger	police_van	prayer_rug
prison	hockey_puck	punching_bag	purse
radio	radio_telescope	rain_barrel	fishing_casting_reel
restaurant	rugby_ball	safe	scabbard
schooner	CRT_monitor	seat_belt	shoe_store
shoji_screen_or_room_divider	balaclava_ski_mask	slide_rule	sliding_door
slot_machine	snorkel	keyboard_space_bar	spatula
motorboat	spider_web	spindle	stage

steam_locomotive	through_arch_bridge	steel_drum	stethoscope
stone_wall	tram	stretcher	stupa
submarine	sundial	sunglasses	sunscreen
suspension_bridge	swing	tape_player	television
thatched_roof	threshing_machine	throne	tile_roof
tobacco_shop	toilet_seat	torch	totem_pole
toy_store	trimaran	triumphal_arch	trombone
turnstile	typewriter_keyboard	vaulted_or_arched_ceiling	velvet_fabric
vestment	viaduct	sink	whiskey_jug
whistle	window_screen	window_shade	airplane_wing
wool	split-rail_fence	shipwreck	sailboat
yurt	website	crossword	dust_jacket
menu	plate	guacamole	trifle
baguette	cabbage	broccoli	spaghetti_squash
acorn_squash	butternut_squash	cardoon	mushroom
Granny_Smith_apple	jackfruit	cherimoya_(custard_apple)	pomegranate
hay	carbonara	chocolate_syrup	dough
meatloaf	pot_pie	red_wine	espresso
tea_cup	eggnog	mountain	bubble
cliff	coral_reef	geyser	lakeshore
promontory	sandbar	beach	valley
volcano	baseball_player	bridegroom	scuba_diver
rapeseed	daisy	yellow_lady's_slipper	corn
acorn	rose_hip	horse_chestnut_seed	coral_fungus
gyromitra	stinkhorn_mushroom	earth_star_fungus	hen_of_the_woods_mushroom
bolete	corn_cob		

Table 4. **Extra categories from ImageNet-1K.**

Bible	pirate_flag	bookmark	bow_(weapon)
bubble_gum	elevator_car	chocolate_mousse	compass
corkboard	cougar	cream_pitcher	cylinder
dollar	dolphin	eyepatch	fruit_juice
golf_club	handcuff	hockey_stick	popsicle
pan_(metal_container)	pew_(church_bench)	piggy_bank	pistol
road_map	satchel	sawhorse	shawl
sparkler_(fireworks)	spider	string_cheese	Tabasco_sauce
turtleneck_(clothing)	violin	waffle_iron	whistle
wind_chime	headstall_(for_horses)	fishing_rod	coat_hanger
clasp	crab_(animal)	flamingo	stirrup
machine_gun	pin_(non_jewelry)	spear	drumstick
cornet	bottle_opener	easel	dumbbell
garden_hose	money	saddle_(on_an_animal)	garbage
windshield_wiper	needle	liquor	bamboo
armor	pretzel	tongs	ski_pole
frog	hairpin	tripod	flagpole
hose	belt_buckle	streetlight	coleslaw
antenna	hook	Lego	thumbtack
coatrack	plow_(farm_equipment)	vinegar	strap
poker_(fire_stirring_tool)	cufflink	chopstick	salad
dragonfly	musical_instrument	sharpener	bat_(animal)
lanyard	mat_(gym_equipment)	gargoyle	underdrawers
paperback_book	razorblade	earring	sword
shovel	turkey_(food)	ambulance	pencil
weathervane	trampoline	applesauce	jam
ski	tray	tissue_paper	lamppost
clipboard	router_(computer_equipment)	battery	lollipop
crayon	latch	fig_(fruit)	sunglasses
toothpick	business_card	padlock	asparagus
shot_glass	sled	key	bolt
pipe	steering_wheel	deck_chair	green_bean
pouch	telephone_pole	fire_hose	ladle
pliers	hair_curler	handle	screwdriver
dining_table	cart	oar	wolf
envelope	legume	shopping_cart	trench_coat

Table 5. **Categories of ChatGPT designed prompts.**

Example of using ChatGPT to design prompts

Human Input:

```
Please write 20 prompts for stable diffusion to generate images of cream pitcher.
Each prompt needs to be different, the focus of the description should be the different attributes
of the category, such as if the category is animal, the prompts should describe the animal's color,
size, age, species, etc., if it is food, the prompts should describe the color of food, brand,
size, freshness, packaging type, packaging color, etc.
Each prompt should ensure that there is only one entity in the generated image. The prompt should
start with words like "a photo of ", "a real photo of", "an image of ", "a photography of ", "an
illustration of ", etc.
The output result should not contain anything else, just a prompt list in the format:
1. promptA
2. promptB
3. promptC
...
The first sentence of the output is "this is the list:".
```

ChatGPT Response:

```
This is the list:
1. A photo of a small cream pitcher with a glossy white finish.
2. An image of a vintage cream pitcher with delicate floral patterns.
3. A real photo of a large cream pitcher made of fine bone china.
4. A photography of a modern cream pitcher with a sleek silver handle.
5. An illustration of a cream pitcher in a vibrant shade of turquoise.
6. A photo of a small cream pitcher with a rustic, weathered appearance.
7. An image of a cream pitcher adorned with intricate gold filigree.
8. A real photo of a tall cream pitcher with a matte black finish.
9. A photography of a cream pitcher in a classic white color with a scalloped edge.
10. An illustration of a cream pitcher shaped like a whimsical cow.
11. A photo of a vintage cream pitcher with a charming polka dot design.
12. An image of a cream pitcher made of clear glass with a frosted bottom.
13. A real photo of a cream pitcher in a unique hourglass shape.
14. A photography of a cream pitcher with a playful print of colorful flowers.
15. An illustration of a cream pitcher with a hand-painted blue and white pattern.
16. A photo of a small cream pitcher made of delicate porcelain.
17. An image of a cream pitcher with a textured surface resembling tree bark.
18. A real photo of a cream pitcher in a vibrant shade of coral pink.
19. A photography of a cream pitcher with a gold-trimmed spout and handle.
20. An illustration of a cream pitcher shaped like a graceful swan.
```

Figure 3. **Example of using ChatGPT to design prompts.**

A photo of a copper tray with a hammered texture, in a white background

A real photo of a large tray made of clear acrylic material, in a white background

A photo of a large tray with vibrant floral designs, in a white background

An image of a wooden tray with intricate carvings, in a white background

A photo of a small ceramic tray in a vibrant turquoise color, in a white background

A photography of a ceramic tray with colorful geometric patterns, in a white background

An illustration of a tray made of recycled materials, in a white background

A photography of a small porcelain tray adorned with intricate blue and white designs, in a white background

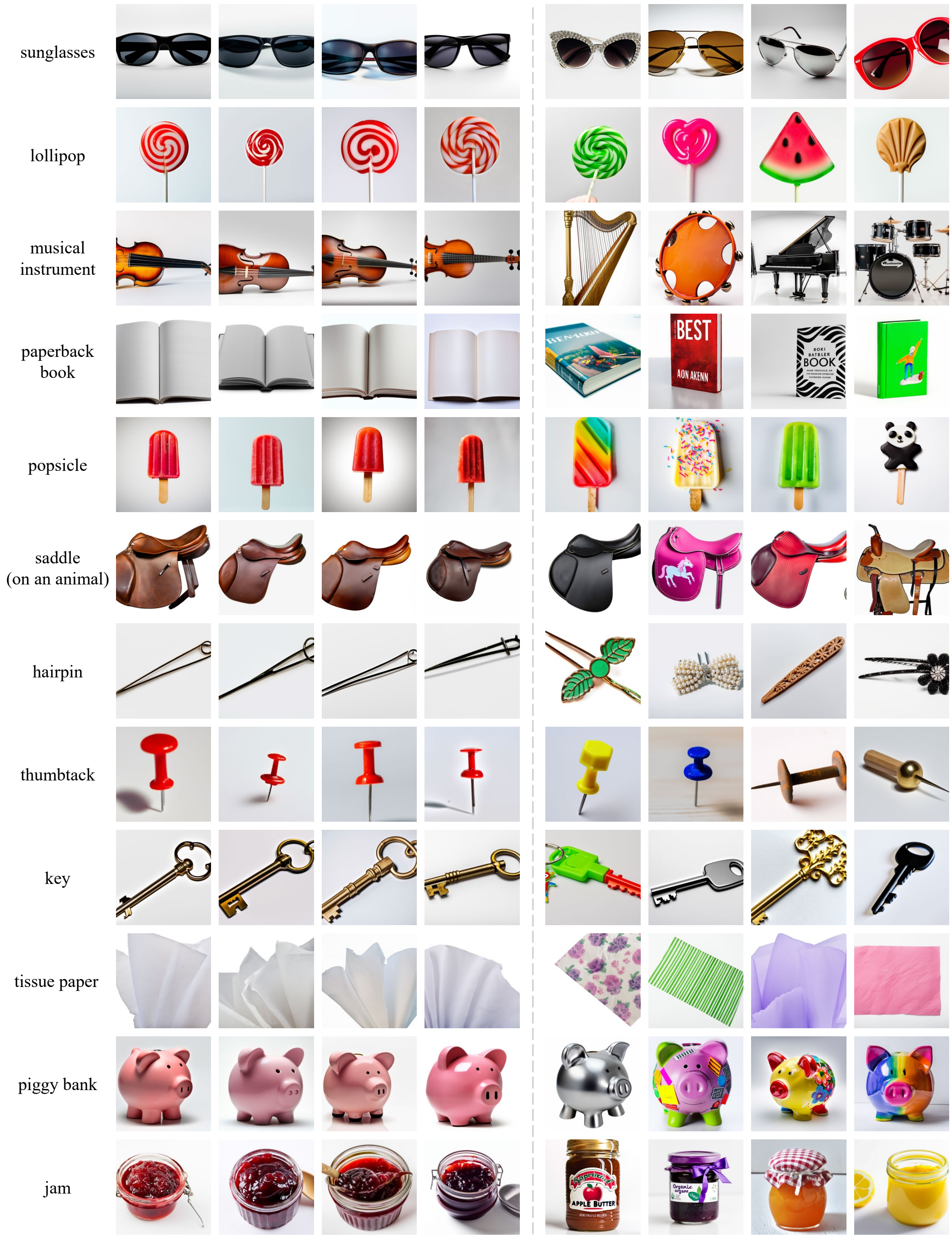
A photography of a tray made of bamboo with a natural brown color, in a white background

A real photo of a crystal tray with sparkling facets, in a white background

A photo of a large tray made of marble with white veins, in a white background

An illustration of a gold tray with a mirrored bottom, in a white background

Figure 4. **Examples of ChatGPT designed prompts and corresponding generative images.** Images generated from ChatGPT designed prompts have diverse textures, styles, patterns, etc.



Images of manually designed prompts.

Images of ChatGPT designed prompts.

Figure 5. **Examples of generative data using different prompts.** By using prompts designed by ChatGPT, the diversity of generative images in terms of shapes, textures, etc. can be significantly improved.

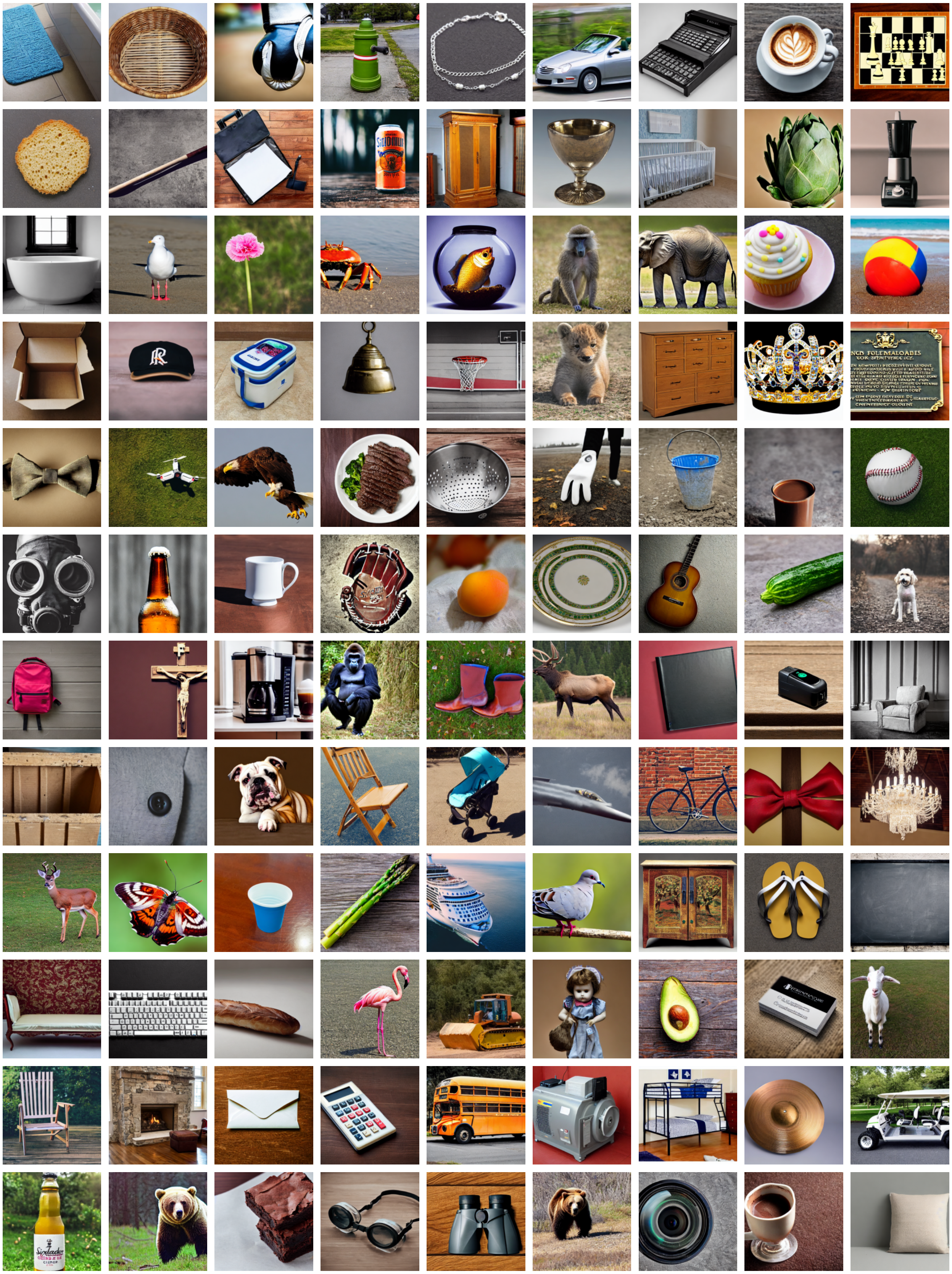


Figure 6. **Examples from Stable Diffusion.** The samples generated by different generative models vary, even within the same category.

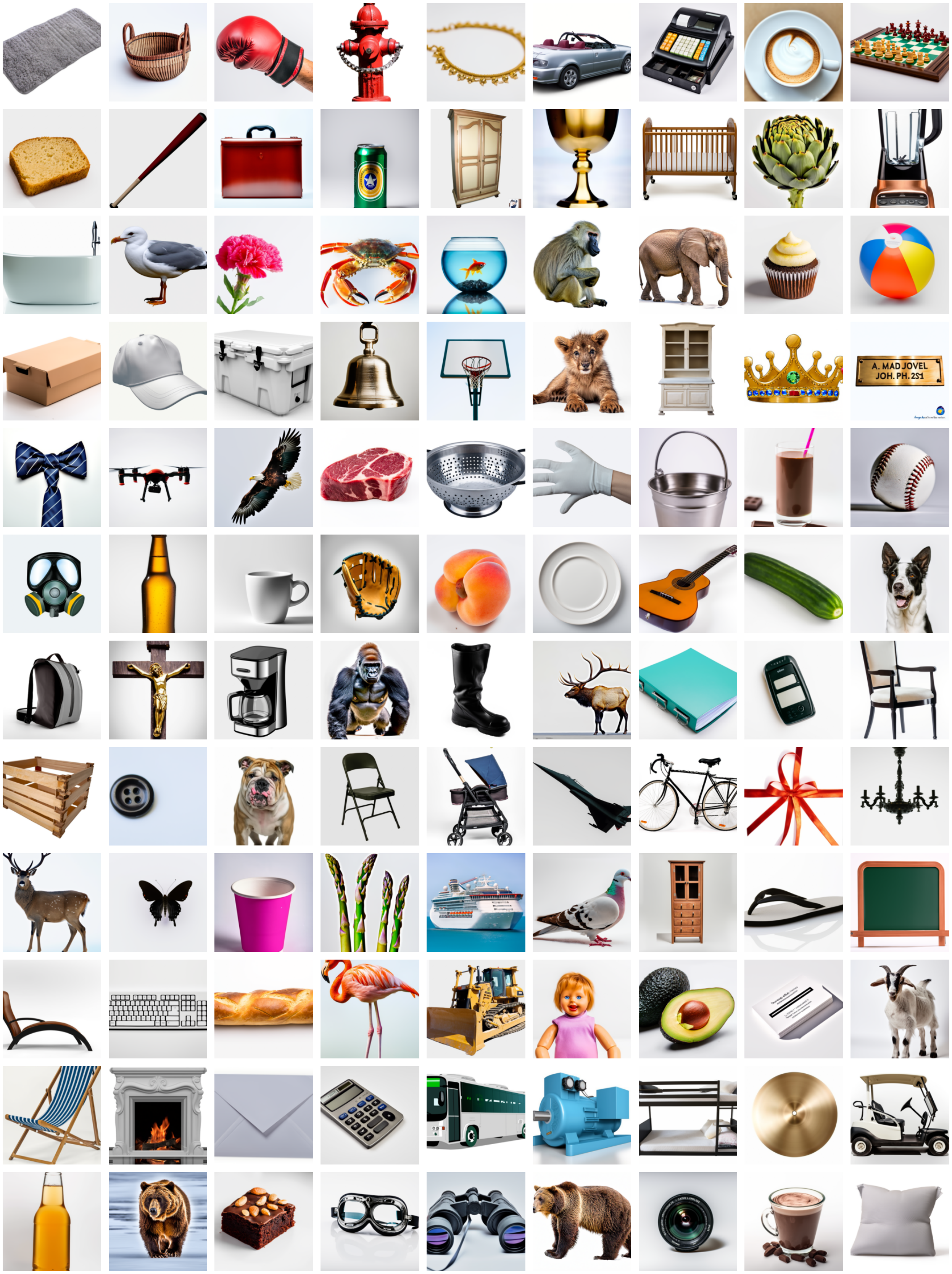


Figure 7. **Examples from DeepFloyd-IF.** The samples generated by different generative models vary, even within the same category.

Figure 8. **Examples of different annotation strategies.** Masks generated by max CLIP tend to be incomplete, while our proposed SAM-bg is able to produce more refined and complete masks when processing images with multiple categories.

Figure 9. **Examples of augmented data.** The use of instance augmentation strategies helps alleviate the limitation in relatively simple scenes of generative data and improves the efficiency of model learning on the generative data.