Conjuring Positive Pairs for Efficient Unification of Representation Learning and Image Synthesis

Supplementary Material

A. Implementation Details

Architecture details. Before the contrastive loss is applied in Sorcen, the output of the student branch is passed through a simple MLP projector of 2 layers and an MLP predictor of another 2 layers. The teacher branch is composed of a smoothed version of the student branch, including the encoder and the projector which are updated via EMA every training step. The teacher branch does not include any predictor. Reconstruction pretraining exclusively follows the student branch and includes an additional Decoder to perform the semantic reconstruction. For all experiments we use $\lambda = 0.1$, $K = 15$ and a projector size of 512 according to analysis provided in Section G.

Evaluation setup. We apply 20 steps to generate images for generative evaluation and all discriminative tasks are conducted by globally averaging the features generated by the student encoder. Teacher encoder, projectors and predictor are completely discarded on inference, using the Student encoder for discriminative tasks and the decoder for generative tasks, unless otherwise stated. To ensure reproducibility and provide a comprehensive understanding of our experimental setup, we detail the implementation specifics in this section. Tables A.1 to A.3 summarize the key hyperparameter settings used for pre-training, linear probing, and few-shot learning experiments, respectively.

config	value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.95
batch size	4096
learning rate schedule	cosine decay
warmup epochs	40
echo warmup epochs	40
training epochs	1600
gradient clip	3.0
label smoothing	0.1
dropout	0.5
masking ratio min	0.5
masking ratio max	1.0
masking ratio mode	0.55
masking ratio std	0.25

Table A.1. Pre-training Setting.

config	value
optimizer	LARS
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	4096
learning rate schedule	cosine decay
warmup epochs	0
training epochs	90
augmentation	RandomResizedCrop

Table A.2. Linear Probing Setting.

config	value
optimizer	LARS
base learning rate	1.0
weight decay	0.0
optimizer momentum	0.9
batch size	16
learning rate schedule	cosine decay
warmup epochs	0
training epochs	10
augmentation	RandomResizedCrop

Table A.3. Few-shot Setting.

B. MAGE Multi-setup Comparison

Table B.4 presents numerical results comparing MAGE and Sorcen, as visualized in Figure 5. As discussed, Sorcen improves upon the state-of-the-art in unified learning, surpassed only by specific MAGE versions that lack balanced performance in both discriminative and generative tasks, thus losing their unified model advantage.

C. Additional Transfer Learning Results

In Tables C.5 and C.6 we further show the transfer learning performance of Sorcen in 8 and 4 shot setups respectively. While reducing the shots, the same behaviour maintains. Sorcen firmly beats MAGE on average for both setups.

D. Precomputed MAGE

In Table D.7 we compare Sorcen with a modified version of MAGE that does not include the VQGAN inference during training. Therefore, MAGE is not able to apply any of the augmentations that it relies on. Both approaches are trained

Method	Epochs	FID	Linear
MAGE	800	11.60	73.30
MAGE	1600	11.10	74.70
MAGE$_{\text{wa}}$	1600	8.67	70.50
MAGE-C (1.0)	800†	14.10	75.00
MAGE-C (0.6)	800†	27.00	77.10
MAGE-C (0.6)	1600†	31.77	78.20
Sorcen	800	10.30	74.28
Sorcen	1600	9.61	75.10

Table B.4. Results on IN1k of different configurations of MAGE and Sorcen for 800 and 1600 pre-training epochs. † indicates the need for two passes of the VQGAN for each training step.

with the same precomputed IN200 dataset for 800 epochs and evaluated on IN200 validation and IN1k validation for 25-shot linear probing accuracy. As can be seen, MAGE falls further behind Sorcen as it was thought to be trained using, at least, Random Resized Crop augmentation which increases the diversity of the samples. This is also hinted in its Weak Augmentation setup (Table B.4), where it suffers a drastic drop on discriminative performance. Sorcen has been carefully designed to train on precomputed tokens and its Echo contrast provides required diversity to the framework. This diversity makes it surpass MAGE on IN1k validation set, which includes 800 new classes for the models.

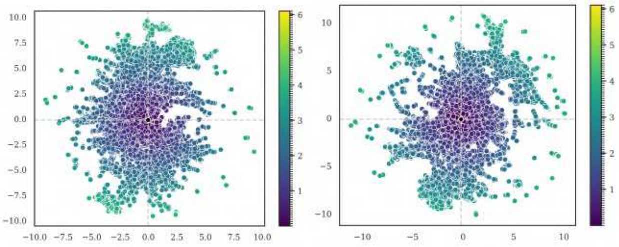
E. Disk Efficient Comparison

During training, Sorcen does not require the VQGAN tokenizer. Every image is preprocessed beforehand and we work with the processed tokens. While our main goal consists in removing the overhead provided by the VQGAN [22], we also reduce considerably the required disk space compared to the rest of the SoTA models. In Table E.8, we compare Sorcen with disk-efficiency-focused models. To the best of our knowledge, Sorcen is the single disk efficient method with SoTA generation capacities. Using just the 0.29% of the original dataset size, it outperforms previous disk efficient strategies, SeiT [48] and competes with SeiT++ [35], which uses dedicated token augmentations. Our approach, using less than a third of the space used by SeiT family [35, 48], manages to produce useful features that work for both discriminative and generative tasks, something impossible for SeiT family, which exclusively work on discrimination. Even if we do not focus on disk efficient learning, Sorcen opens an interesting line where 0.39GB of preprocessed data are enough to provide SoTA discriminative and generative results. This dataset will be released upon acceptance.

F. Hybrid Space Visualization

Following other works in the literature [23, 24], we use UMAP [42] to visualize low-dimensional representations of the latent space of the model. Thanks to this visualization, we can gain insight into how elements from different semantic classes are distributed in the representation space.

Figure F.1 shows the UMAP representation of the whole IN1k validation set for both MAGE and Sorcen. According to this visualization, it is easy to see that, while the MAGE space resembles a "blob", with almost no distinguishable clusters (apart from a few outliers), the space achieved by Sorcen presents easier-to-separate groups while preserving the dense nature of the MAGE space. This duality explains why Sorcen is able to excel in both discriminative and generative tasks by leveraging a common hybrid space.



(a) MAGE (b) Sorcen

Figure F.1. Latent space visualization (UMAP) of MAGE and Sorcen embeddings for the IN1k validation set. Each point represents an image, colored according to its distance (in standard deviations) from the mean embedding (black dot).

To further investigate the semantic organization within the latent space, we now visualize UMAP representations, color-coded by semantic class. Figure F.2 showcases these visualizations for a selection of representative classes from the IN1k training set, comparing both MAGE and Sorcen models. This class-based coloring allows us to examine the degree of separability between different semantic categories in each model's latent space, particularly for fine-grained categories.

Across the visualized classes, a consistent trend is observed: Sorcen's latent space exhibits markedly superior class separability compared to MAGE. While both models display some intra-class dispersion and inter-class mixing, this phenomenon is significantly more pronounced in MAGE. In Sorcen's representation, semantic classes are notably more distinct and less intermingled.

This enhanced class separability in Sorcen's latent space provides further insight into its hybrid capabilities. The formation of more well-defined and separated class clusters likely underpins its stronger discriminative performance. Simultaneously, the preservation of an overall dense latent space, as demonstrated previously, still accommodates generative flexibility. Consequently, these class-based UMAP

Method	Caltech	UCF101	Flowers	Pets	Sun	EuroSAT	DTD	Avg.
MAGE	83.94	46.37	**68.33**	48.79	43.38	40.01	**40.13**	52.99
Sorcen	**85.31**	**50.70**	67.37	**50.89**	**44.65**	**46.70**	39.60	**55.03**

Table C.5. **Transfer learning results (top-1 accuracy) for different datasets under 8-shot settings.** Last column contains the average across datasets.

Method	Caltech	UCF101	Flowers	Pets	Sun	EuroSAT	DTD	Avg.
MAGE	**71.72**	29.71	**46.00**	24.69	31.36	18.75	19.92	34.59
Sorcen	69.20	**36.35**	43.89	**24.91**	**33.48**	**25.01**	**21.99**	**36.40**

Table C.6. **Transfer learning results (top-1 accuracy) for different datasets under 4-shot settings.** Last column contains the average across datasets.

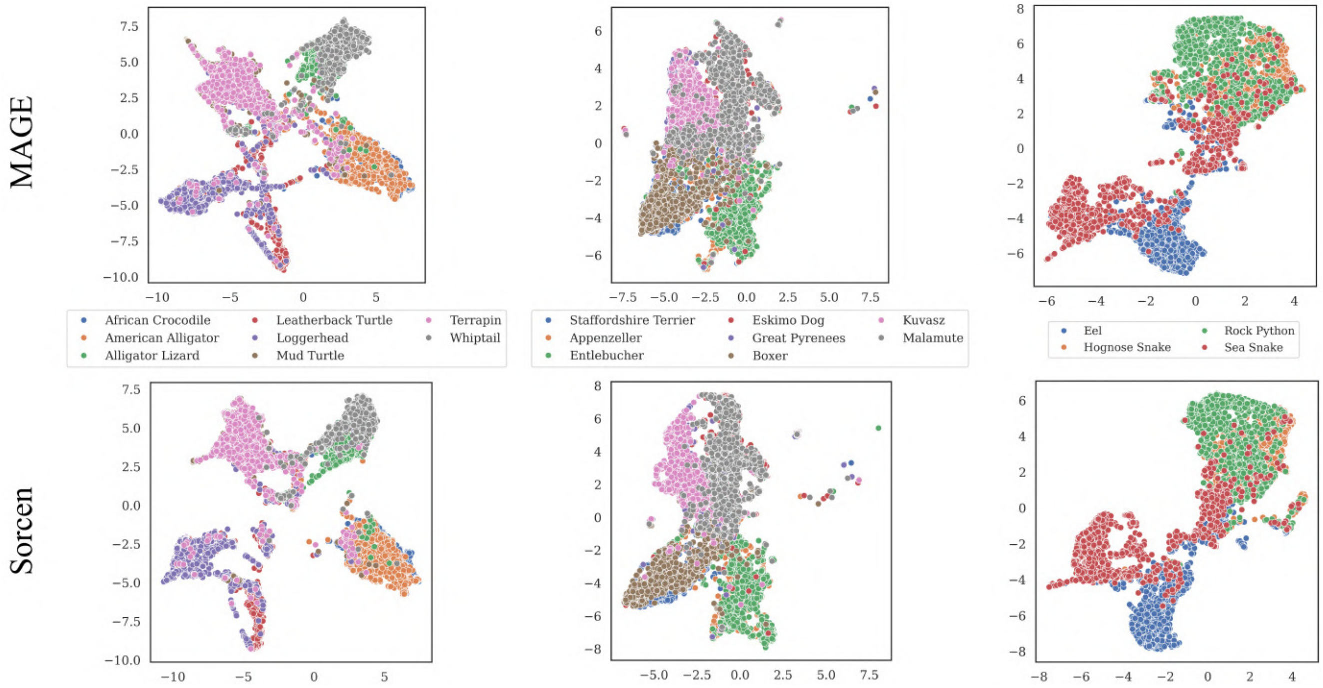


Figure F.2. Class-specific UMAP visualizations of latent embeddings from MAGE (top row) and Sorcen (bottom row) for the IN1k training set. Points are colored by semantic class to illustrate class separation within each model's latent space.

Method	IN1k val †	IN200 val
MAGE$_{precomp}$	33.25	69.80
Sorcen	36.15	74.10

Table D.7. **Results on precomputed token setups.** We highlight IN1k val with a † as it includes 800 new classes for the models, acting as new domain and showing the generalization capabilities of the models.

visualizations strongly suggest that Sorcen achieves a more semantically structured hybrid space. This structure effectively balances discriminative and generative representation learning, contrasting with MAGE's comparatively less organized latent space.

G. Impact of Hyperparameters

In this section, we delve into the impact of key hyperparameters on the performance of our Sorcen model. Through a series of ablation studies, we analyze the influence of the contrastive loss coefficient λ, the K parameter in top-k logit sampling, and the projector size. Due to computational constraints, we run these experiments on IN200 dataset for 200 epochs (1024 batch size) and evaluate them on few-shot regime with 25 shots per class.

Method	# of images	Storage size	Top-1	FID
MAGE [38]	1,281k	140GB (100%)	74.7	11.1
Uniform†	512k	54.6 GB (39%)	74.0	-
C-score† [33]	512k	53.3 GB (38%)	73.3	-
JPEG 5†	1,281k	11.0 GB (8%)	74.6	-
SeiT [48]	1,281k	1.4 GB (1%)	74.0	-
SeiT++ [35]	1,281k	1.4 GB (1%)	**77.8**	-
Sorcen	1,281k	**0.39 GB (0.29%)**	75.1	**9.61**

Table E.8. **Storage-efficient evaluation.** Linear accuracy and FID are provided. Sorcen is the single storage-efficient method with image generation capabilities. † results reported in [48].

	0.01	0.1	1.0
Top1	66.77	69.62	47.77
FID	22.60	22.41	46.82

Table G.9. Study of the impact of the value of the contrastive loss coefficient λ in accuracy (linear probing) and FID in IN200.

Contrastive loss coefficient. We ablate the contrastive loss coefficient in Table G.9. Consistent with prior work [32, 38, 61], reconstruction remains the dominant loss. However, reducing it below 0.1 inhibits the advantages provided by our Echo contrast, reducing the overall performance of the system. However, our ablation in Table G.9 reveals that the contrastive loss is essential. Reducing the coefficient to 0.01 leads to a decrease in Top-1 accuracy to 66.77% and a slight increase in FID to 22.60. In contrast, increasing the coefficient to 1.0 drastically degrades both metrics, with Top-1 accuracy falling to 47.77% and FID increasing significantly to 46.82. This highlights that a balanced coefficient of 0.1 effectively balances discriminative and generative performance, achieving a good trade-off in Top-1 accuracy and a low FID score.

K logit sampling. Sorcen exhibits remarkable robustness to the K parameter in top-k logit sampling as proved in Table G.10. While larger K values introduce a potential risk of sampling less accurate tokens, the skewed nature of the logits produced by the decoder mitigates negative impacts. Conversely, smaller K reduces this risk but might also limit Echo sample diversity. Empirical results show negligible performance variation across K=5, 15, and 30, with Top-1 accuracy hovering around 69.6% and FID consistently low (approximately 22.4-22.8). We opted for K=15 as a balanced choice, mitigating inaccuracy risks while maintaining sufficient diversity in Echo samples.

Projector size. Sorcen achieves effective contrastive learning without requiring an excessively large projector. Table G.11 shows that increasing projector size beyond 512 dimensions does not yield further performance gains. In fact, expanding from 256 to 512 improves Top-1 accuracy (69.36% to 69.62%) and FID (22.74 to 22.41), indicating that 512 dimensions sufficiently capture relevant semantic information for contrast. However, further enlargement to 1024 and 2048 dimensions provides no significant advantage and even slightly degrades performance, suggesting diminishing returns and potential introduction of unnecessary model complexity beyond a projector size of 512.

	5	15	30
Top1	69.66	69.62	69,60
FID	22.69	22.41	22.77

Table G.10. Study of the impact of the number of tokens, K, considered to create the Echoes. Results correspond to IN200 pretraining and evaluation: linear probing and FID.

	256	512	1024	2048
Top1	69.36	69.62	69.55	69.45
FID	22.74	22.41	22.52	22.19

Table G.11. Impact of the size of the projector in top-1 accuracy (linear probing) and FID in IN200.

H. Extended Robustness Evaluation

In Table H.12 we extend our main papers k-NN robustness results to ImageNet-C, ImageNet-A and ObjectNet datasets. On **IN-C**, Sorcen outperforms MAGE at Severity 4 and 5, the hardest levels, with scores of **22.15 vs. 21.96** (S4), and **14.42 vs. 13.69** (S5), averaged over 19 corruptions. On **IN-A**, Sorcen achieves **5.47**, vs MAGE's **4.19**. On **ObjectNet** (113 **out-of-distribution** IN classes), Sorcen and MAGE score **13.52 vs. 10.31**.

	IN-C Sev. 4	IN-C Sev. 5	IN-A	ObjectNet
MAGE	21.96	13.69	4.19	10.31
Sorcen	22.15	14.42	5.47	13.52

Table H.12. Extended robustness. IN-C shows the mean of all 19 corruption types.

I. Conditional Generation

While Sorcen is completely focused unsupervised pretraining and generation, we extend the generative evaluation results by adding a conditioned decoder. For both Sorcen and MAGE, we freeze the pretrained encoder and retrain the decoder by replacing the extra token by a CLIP token. This token is computed using the common ”An image of a

[CLASS]" template and enables the use of external guidance without requiring any additional token. In Table I.13, we can see how both obtain similar performance. While our small setup achieves modest results, we show that, on conditional generation, Sorcen competes with MAGE while being superior in all the rest of the tasks tested.

	FID	IS	PR F1 score
MAGE	7.79	130.88	54.8
Sorcen	7.78	127.66	54.96

Table I.13. Conditional Generation results.

J. Extended ablations

In Table J.14 we extend JSM and L2 loss ablations, proving the value of Sorcens architectural choices on a higher scale setups. L2 loss improves FID, but lowers Top-1 accuracy, backing InfoNCE as a better choice. Equally, Table J.15 further explores the capacity of Sorcen when applied on bigger backbones such as ViT-L.

	Top-1	FID	IS
noJSM	63.18	10.57	85.13
L2 Loss	62.91	9.95	85.91
Sorcen	63.51	10.20	83.98

Table J.14. Extended ablation study on a higher scale setup

	Top-1	FID
MAGE	67.29	21.02
Sorcen	71.89	17.78

Table J.15. Extended ablation study on a ViT-L backbone. Note that these results are computed using IN200 precomputed tokens for 200 epochs.

K. Token-based pretraining on Dense Tasks

Token-based approaches, such as MAGE and Sorcen, inherently underperform on dense tasks (detection, segmentation) due to reduced spatial detail [53]. As can be seen in Table K.16, on MSCOCO, ADE20K, and FoodSeg103, Sorcen matches MAGE on general datasets and outperforms it on the specific one.

	MSCOCO	ADE20K	FoodSeg
MAGE	15.80	29.29	15.88
Sorcen	15.90	29.47	16.82

Table K.16. Extended evaluation on Instance Segmentation downstream task. mAP metric is reported.

L. Qualitative Results

Thanks to the SoTA image generation achieved by Sorcen, we are able to show in this section different generative applications of this model to modify and generate high-quality images. This can be done completely from scratch (unconditioned generation) or from a given part of an image. We also exhibit randomly selected Echoes from Sorcen's pretraining. All the images shown in this section have been generated by Sorcen ViT-B trained for 1600 epochs on IN1k.

Echoes extracted during training. In Figures L.7 and L.8 we display some of the Echoes generated by Sorcen during training.

Unconditioned generation. Figure L.3 contains 63 examples of images randomly generated by Sorcen in an unconditioned way.

Image inpainting/outpainting and reconstruction. Figures L.4 to L.6 contain examples of inpainting, outpainting and reconstruction applied to images from IN1k dataset. Each rows correspond to an image. For reconstruction, we use a random mask of 75% of the image. The first column is the original image, and the next columns contain pairs masked image-generation for three masks.

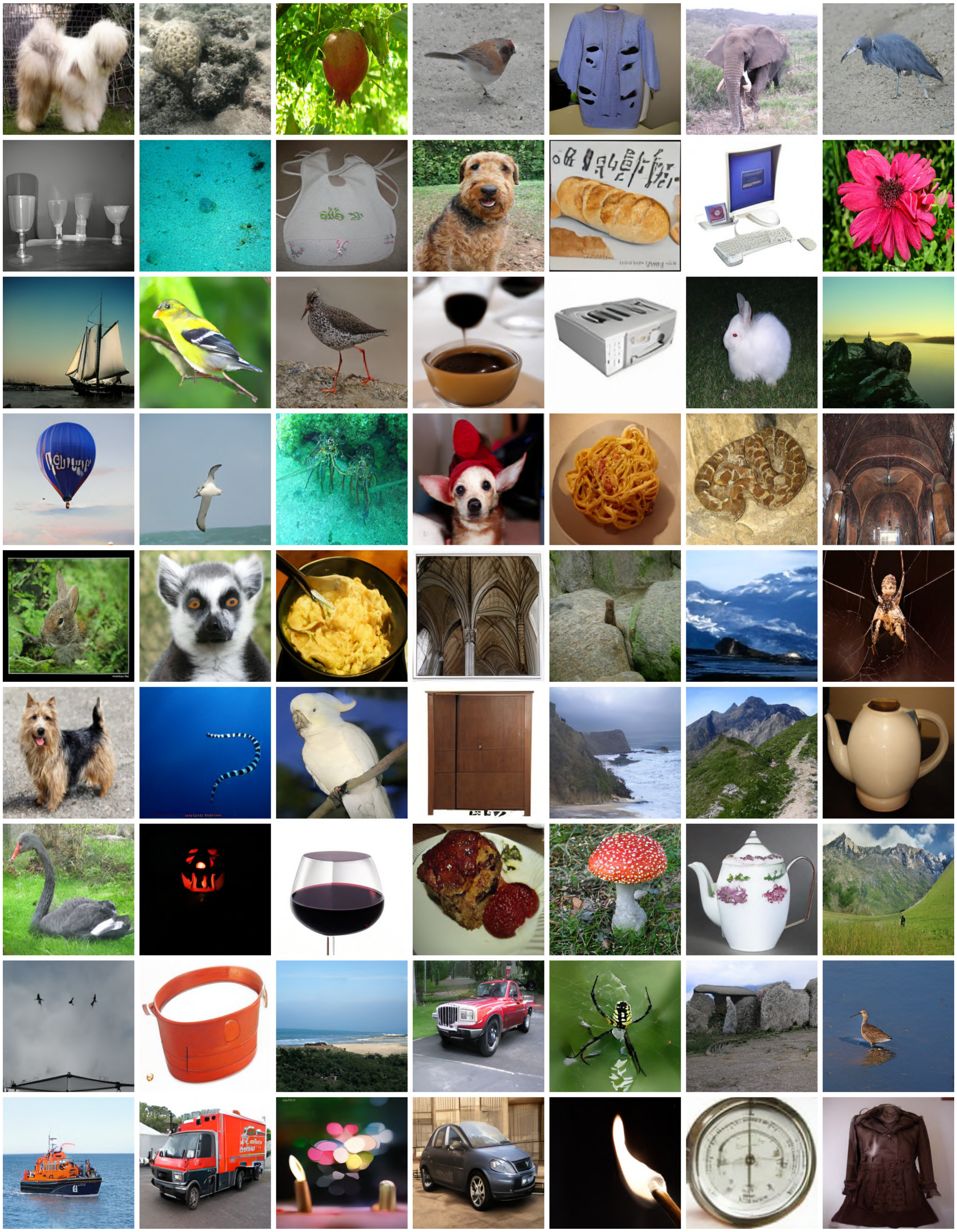


Figure L.3. Images randomly generated by Sorcen in an unconditioned way using a ViT-B.

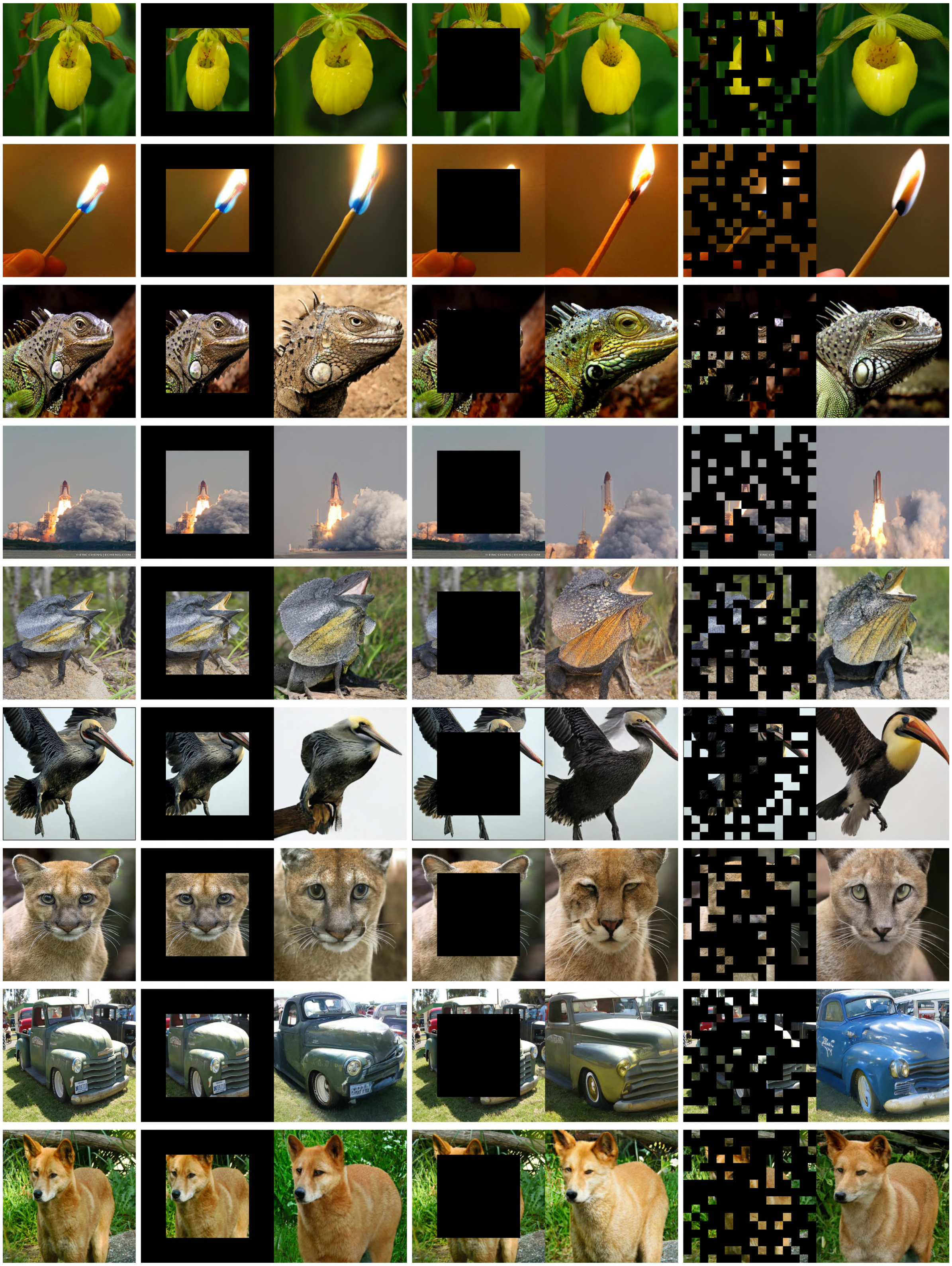


Figure L.4. Examples of image inpainting, outpainting and reconstruction with Sorcen ViT-B. The first column is the original image.

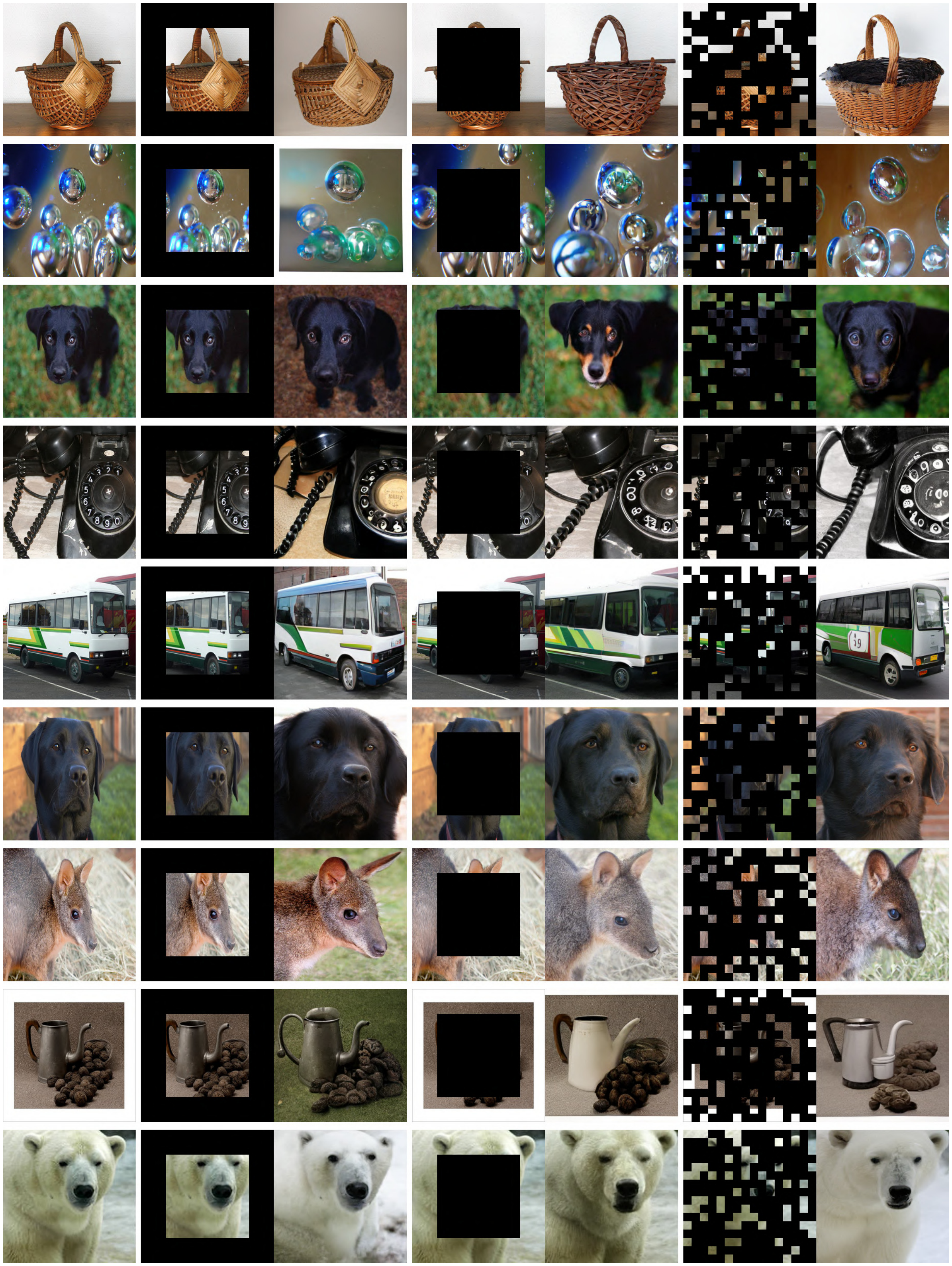


Figure L.5. Examples of image inpainting, outpainting and reconstruction with Sorcen ViT-B. The first column is the original image.

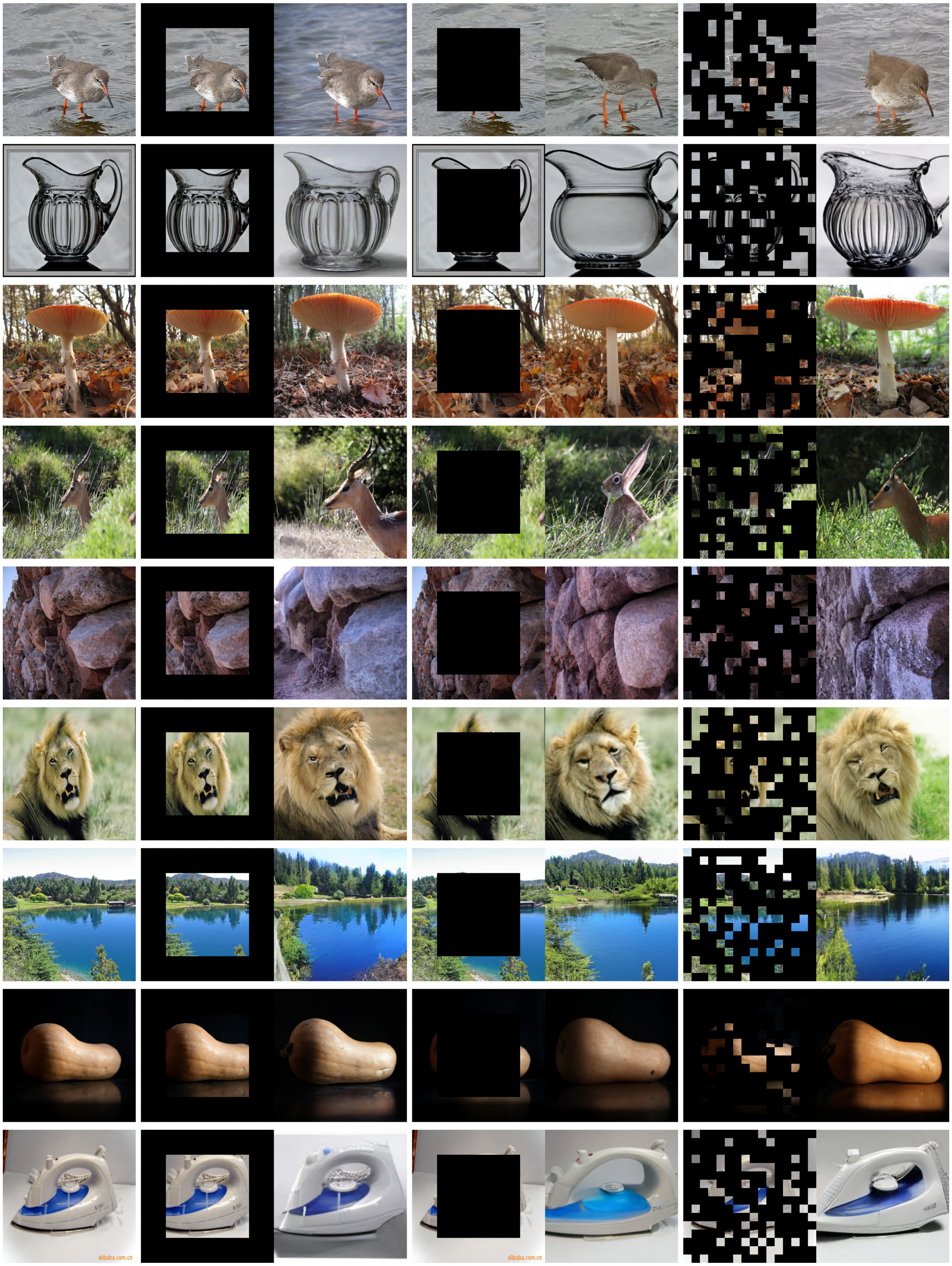


Figure L.6. Examples of image inpainting, outpainting and reconstruction with Sorcen ViT-B. The first column is the original image.

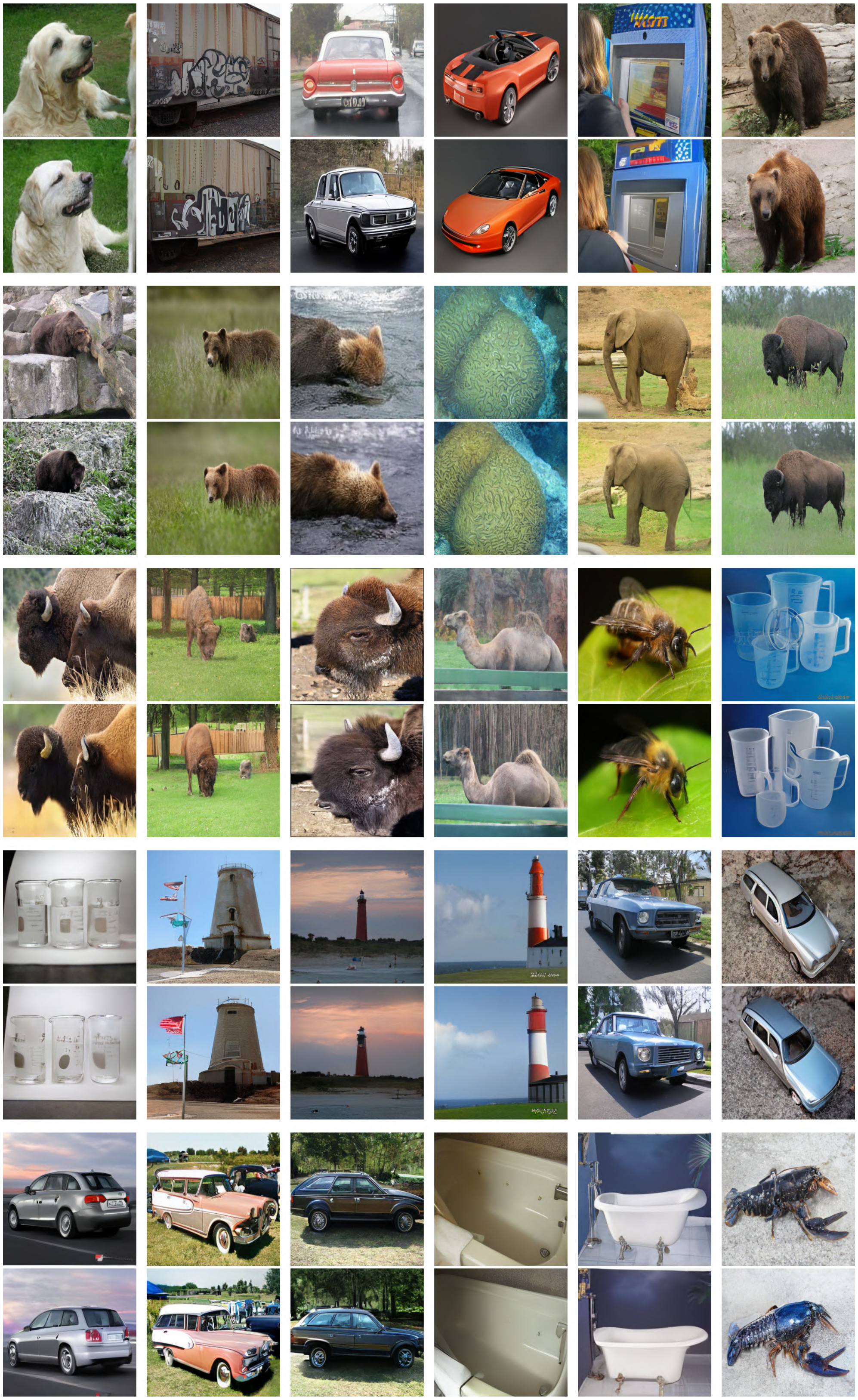


Figure L.7. Uncurated collection of Echo samples generated during Sorcen training for contrasting. For each couple of rows, the top one contains the original image, and the bottom one the Echo.

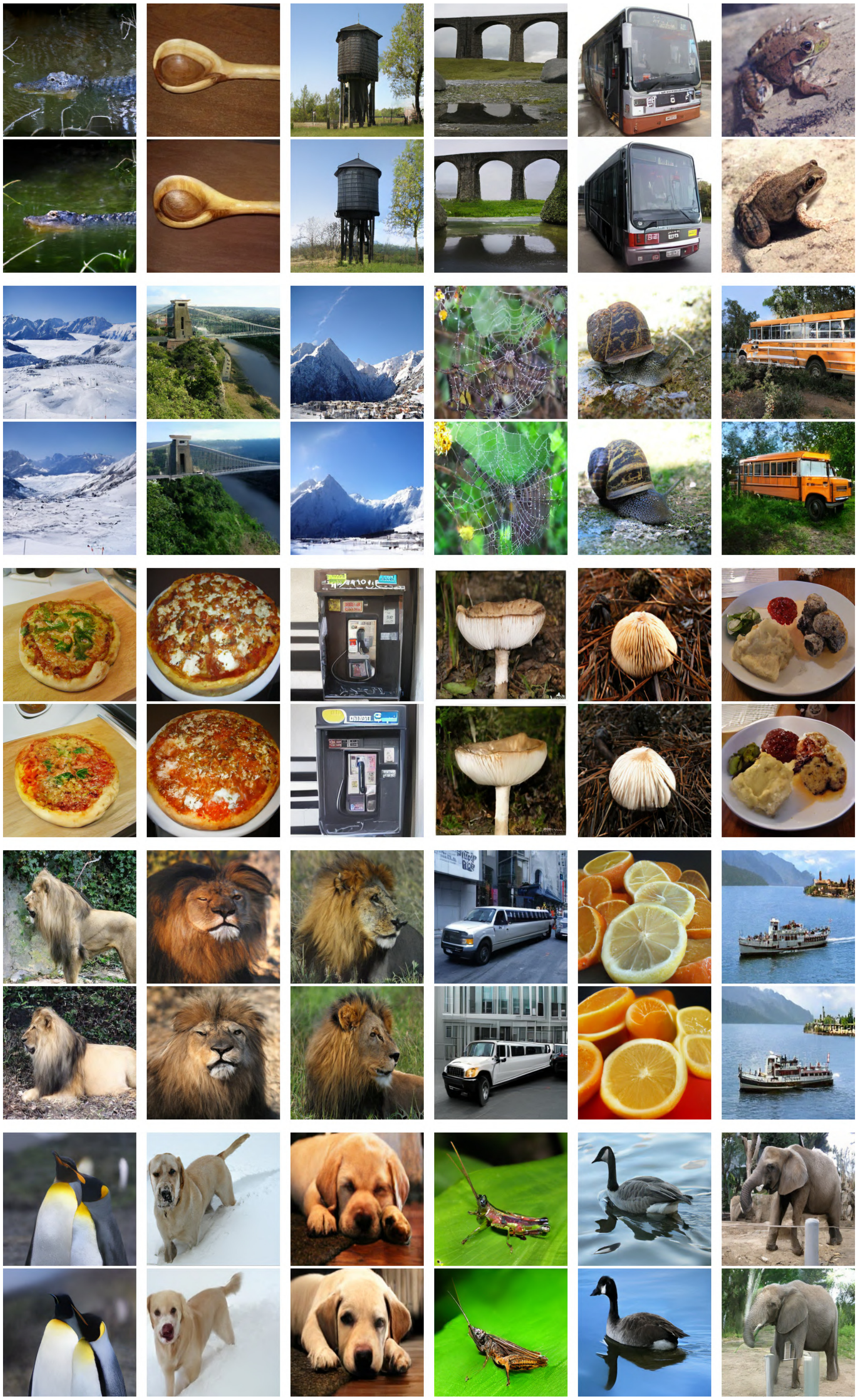


Figure L.8. Uncurated collection of Echo samples generated during Sorcen training for contrasting. For each couple of rows, the top one contains the original image, and the bottom one the Echo.