A Simple Latent Diffusion Approach for Panoptic Segmentation and Mask Inpainting: Supplementary Materials

Wouter Van Gansbeke* and Bert De Brabandere[1]

[1] Segments.ai

We discuss the implementation details in Supplement A, additional results in Supplement B, limitations in Supplement C and the broader impact in Supplement D.

A Implementation Details

Model Card. Our best model is trained for 100k iterations on COCO with mixed precision training on 8 × 40GB NVIDIA A100 GPUs using Google Cloud. We rely on the pretrained Stable Diffusion [16] weights provided by Hugging Face [4]. We also adopt its settings for the noise scheduler. The code is developed in Pytorch [14] and will be made available as well as our models.

Multi-Task Setup. This section provides additional information on the multi-task extension for dense prediction, with minor adaptations. Consider the three fundamental vision tasks: instance segmentation, semantic segmentation and depth prediction. The instance and semantic tasks both utilize the same shallow autoencoder to generate continuous latent codes. Similarly, to compress the depth maps, we rely on the same shallow autoencoder architecture as its segmentation counterpart. We only change the input and output channels to one channel. As COCO does not contain depth annotations, we rely on the predictions from MiDaS [1] to obtain pseudo ground truth. Note that this model predicts relative depth. All tasks use the same set of augmentations and scaling factors, as discussed in the main paper. To enable multi-tasking, we introduce learnable task embedding (786-dimensional) via the cross-attention layers of the UNet. This allows us to query the model for a specific task. Figure S1 visualizes the results for each task by only changing the task embedding. We observe that the model can predict accurate instance, semantic and depth maps for a given image. Finally, given our shallower encoder and task embeddings, a comparison with Marigold [7], a concurrent work on depth estimation, could be insightful.

Mask2Former Baselines. Mask2Former [2] is a specialized segmentation framework that produces excellent results for panoptic segmentation. We follow the training recipe from ViTDet [10] and SAM [8] to leverage plain ViT backbones [3]

* This work was done while the author was at Segments.ai, and the final training run was conducted at INSAIT. The author is now affiliated with Google DeepMind.

Fig. S1: Multi-Task Setup - Qualitative Results. The figure displays the results for several images in the COCO val set. We can query the model for multiple tasks as it has learned their respective task embeddings.

with MAE pretrained weights [5]. Specifically, the model consists of the vision transformer backbone, a shallow neck, and a mask decoder. The latter contains 6 masked attention decoder layers and 128 object queries, following [2]. The loss requires Hungarian matching [9] to handle the permutation invariance of the predictions during training. To report the results, we follow its post-processing strategy to combine the classification and mask branches. We adopt the same augmentations as in the main paper, *i.e.*, square resizing and random horizontal flipping. This baseline strikes a good balance between performance, complexity, and training speed. Additionally, we provide results by relying on the backbone and pretrained weights of DINOv2 [13], as we found this to outperform MAE pretrained weights for a ViT-B backbone. We train the models with a batch size of 32 and a learning rate of $1.5e^{-4}$ for 50k iterations on 8 $\times$ 16GB V100 GPUs.

Evaluation Procedure. Our model produces excellent predictions when only relying on the arg max operator. No additional processing is used for the visualizations (see row 3 in figures S3 and S4). To report the final PQ metric, however, we eliminate noise by thresholding the predictions at 0.5 (after applying softmax) and filtering out segments with an area smaller than 512. These results are shown in the last row of figures S3 and S4. Notice that Mask2Former's training objective does not impose exclusive pixel assignments, hence it needs additional post-processing steps.

Simple Post-Processing. Panoptic segmentation combines instance and semantic segmentation. After efficiently decoding the latents, we will obtain the panoptic mask by starting from the instances. We subsequently take a majority vote using

the predicted semantic mask for each instance. We carry out the following steps and refer to Supplement B for more information on the inference time:

```
def postprocess_panoptic(mask_logits, semantic_logits):
    """
    Convert predictions to panoptic masks.

    Inputs:
        mask_logits: np.array of size [N, H, W]
        semantic_logits: np.array of size [N, H, W]
    Outputs:
        panoptic_seg: np.array of size [H, W]
        segments_to_categories: dict
    """

    panoptic_seg = np.argmax(mask_logits, axis=0)
    semantic_seg = np.argmax(class_logits, axis=0)

    segments_ids = {}
    for segment_id in np.unique(panoptic_seg):
        instance_mask = panoptic_seg == segment_id
        if not_confident_or_small(instance_mask):
            panoptic_seg[instance_mask] = VOID_id
            continue
        counts = np.bincount(semantic_seg[instance_mask]])
        class_id = np.argmax(counts)
        segments_to_categories[segments_id] = class_id

    return panoptic_seg, segments_ids
```

B Additional Results

More Segmentation Results. We show the panoptic segmentation results with 50 timesteps on COCO `val2017` [11] in Figure S2. Additionally, we show (class-agnostic) masks in Figures S3 and S4. The input images are resized to $3 \times 512 \times 512$ during training and the diffusion process acts on latents of size $4 \times 64 \times 64$. To visualize the masks, we assign each segment to a random color. Overall, the model is capable of generating high-quality panoptic masks.

Number of Denoising Steps. Figure S5 displays the results for different timesteps during the denoising process. Longer sampling benefits the generation of details, such as capturing small objects in the background or an object's edges. This approach necessitates 10 - 50 iterations to produce high-quality segmentation masks, which is in line with latent diffusion models for images [16]. Furthermore, as the model was forced to distinguish between different instances during training, it's unlikely that different instances will be grouped during inference. Interestingly, the model iteratively improves the predictions while not reinforcing mistakes during the generative process.

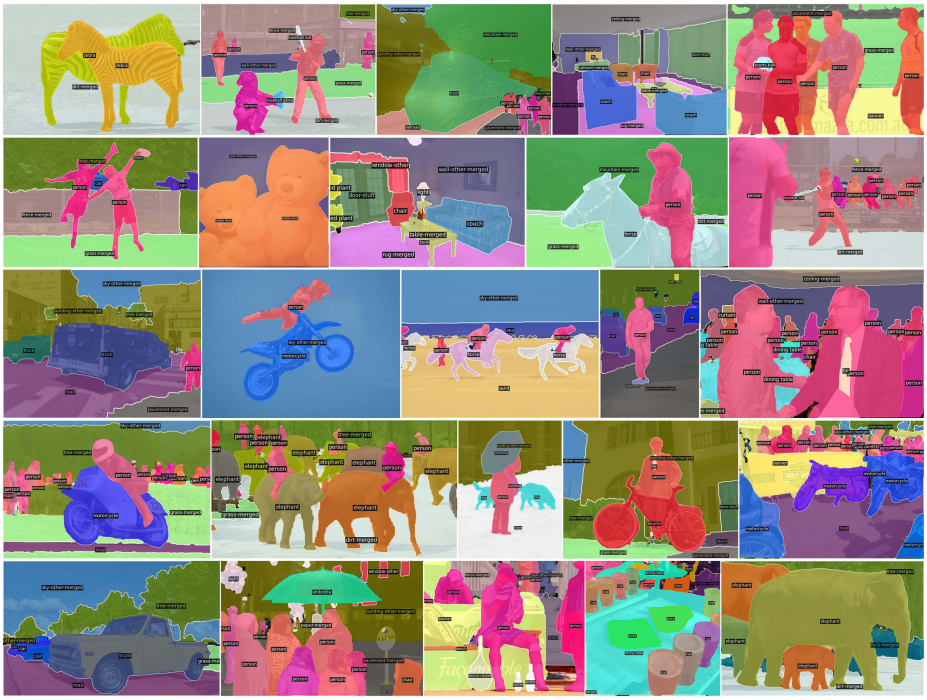

Fig. S2: Panoptic Segmentation - Qualitative Results. The figure displays the panoptic segmentation for several images in the COCO val set.

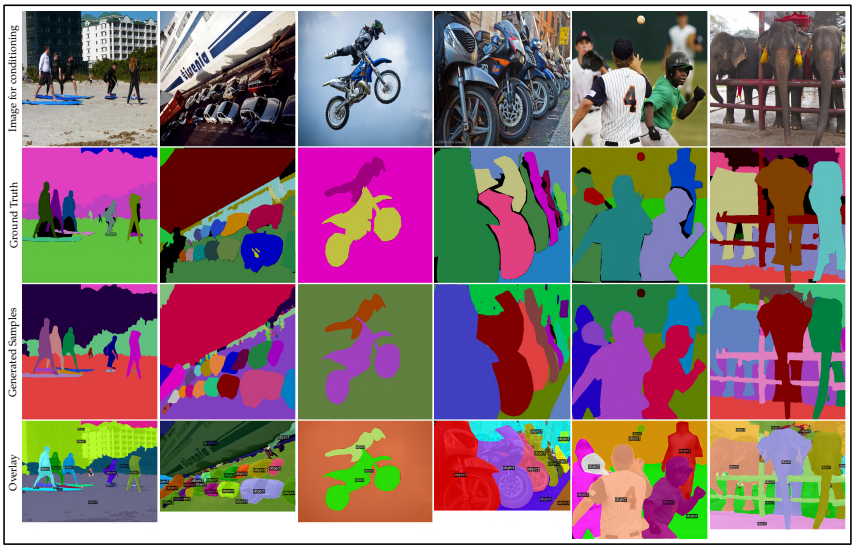

Fig. S3: Examples on COCO (1). The figure displays the generated masks on the COCO val set.

Fig. S4: Examples on COCO (2). The figure displays more generated masks on the COCO val set.

Inference Time. Table S2 provides the inference times for different sampling durations. In comparison, Painter requires approximately 0.5 and 0.7 seconds to post-process an image at a resolution of 448 and 560 respectively on our machine. Our post-processing method is significantly faster, taking up only about 0.024 seconds. Importantly, the performance will vary based on hardware and system specifications. Our relatively simple post-processing is explained in Supplement A (final paragraph). Finally, recent research [18] on Consistency Models looks promising to generate high-quality masks in a single step.

Encoding Panoptic Maps. Table S1 verifies our hypothesis w.r.t. the encoding scheme, as discussed in Sec. 3.1 (main paper). In particular, we test 3 encoding schemes: color (RGB) encoding *vs*. bit encoding *vs*. positional encoding:

- **Colors:** we generate 256 equidistant colors within the RGB space.
- **Bits:** we employ 8 channels to represent integers from $[0, 255]$ using bits.
- **Positional:** we map integers from $[0, 255]$ to an 8-dimensional embedding following [12].

The mIoU and class-agnostic PQ are adopted to measure the reconstruction quality of the autoencoder. We hypothesize that the mapping from color to instance is sub-optimal as this scheme is sensitive to the chosen color palette (89.9 *vs*. 89.1% PQ). In contrast, bit encoding is a general way to represent discrete panoptic maps, which also outperforms positional encoding (89.9 *vs*. 88.2% PQ).

Fig. S5: Results for different timesteps. The figure visualizes the image-conditioned samples for the timesteps 1, 5, 10, 20, and 50 in the diffusion process. Longer sampling is required to capture more details, which is beneficial for complex scenes (*e.g.*, cars in the background in column 4).

Tokenizers and Component Analysis. Table S3 shows that image tokenizers with more semantically meaningful image features can boost the results. In addition, we show the impact of employing different schedulers and an exponential moving average of the model weights. Note that the results are provided with 50 timesteps during inference. All components further enhance the performance of LDMSeg. To summarize, our best results are obtained with a ViT-B [3] architecture and DINOv2 [13] weights as the image encoder, the DDPM scheduler [6] and an exponential moving average of the model weights during training (weight of 0.999).

Loss Weights. Finally, we note that lowering the loss for small timesteps (*e.g.*, $j < 25\%$) is not crucial, but speeds-up training by 0.3 to 0.5% PQ. We aim to remove this in future work.

Table S1: Encoding. Reconstruction quality for different encoding schemes.

Encoding	**mIoU**	**PQ [%]**
bit encoding	97.3	89.9
color encoding	97.0	89.1
positional encoding	96.7	88.2

Table S2: Inference time. We report the average time to generate a single panoptic mask on COCO with a 4090 GPU. The table provides the results for various denoising steps.

	Class-agn. Panoptic Seg.			**Sem. Seg.**	**Panoptic Seg.**			
# Iters	**PQ [%]**	**SQ [%]**	**RQ [%]**	**mIoU [%]**	**PQ [%]**	**SQ [%]**	**RQ [%]**	**Time [s]**
1	8.4	76.0	11.1	18.2	8.1	68.9	10.8	0.115
2	35.5	83.9	42.3	21.3	19.8	78.4	24.8	0.160
3	42.4	84.3	50.4	42.1	35.5	79.6	43.8	0.207
4	45.5	84.2	54.0	51.8	39.3	80.3	48.2	0.259
5	47.3	84.1	56.2	55.1	41.3	80.6	50.3	0.320
10	50.2	83.5	60.1	58.6	43.4	80.4	52.6	0.575
15	51.0	83.3	61.2	58.8	43.7	81.3	53.0	0.815
20	51.4	83.2	61.8	59.1	44.1	81.2	53.4	1.071
25	51.7	83.1	62.2	59.6	44.3	81.3	53.7	1.336
30	51.8	83.0	62.4	59.5	44.1	81.0	53.7	1.585
40	52.0	82.9	62.7	59.3	44.3	81.1	53.8	2.062
50	51.9	82.9	62.6	59.9	44.3	81.1	53.8	2.548
60	52.2	82.8	62.8	59.3	44.4	81.2	53.7	3.074
70	52.2	82.7	63.1	59.4	44.3	81.1	53.7	3.564
80	52.2	82.6	63.1	59.3	44.3	80.5	53.8	4.024
90	52.2	82.6	63.1	59.5	44.3	80.5	53.7	4.550
100	52.1	82.7	63.1	59.1	44.3	81.2	53.7	5.030
200	52.1	82.5	63.2	59.1	44.3	80.5	53.7	10.050

Table S3: Component Analysis.

Setup	Image Encoder	Scheduler	EMA	PQ [%]
1	SD VAE [16]	DDIM [17]	✗	40.3
2	SD VAE [16]	DDIM [17]	✓	40.6
3	ViT-B/14 [3]	DDIM [17]	✓	43.7
4	ViT-B/14 [3]	DDPM [6]	✓	44.3

C Limitations and Future Work

Undoubtedly, our model has several limitations despite its general design. We discuss two limitations: (*i*) the model can miss small background objects due to the projection to latent space; (*ii*) the model is slower during inference than specialized segmentation models due to the adoption of a diffusion prior. In exchange, our method is simple, general and unlocks out-of-the-box mask inpainting. Moreover, the approach can be extended to a multi-task setting. As we rely on plain diffusion models, new innovations (*e.g.*, architectural, noise scheduler, tokenization, number of inference steps *etc.*) in image generation are directly applicable to the presented framework. Finally, increasing the dataset's size, increasing the latents' resolution, enabling open-vocabulary [15] detection, and including more dense prediction tasks are exciting directions to explore further.

D Broader Impact

The presented approach relies on pretrained weights from Stable Diffusion [16]. Consequently, our model is subject to the same dataset and architectural biases. The user should be aware of these biases and their impact on the generated masks. For instance, these types of (foundation) models can hallucinate content.

References

1. Birkl, R., Wofk, D., Müller, M.: Midas v3. 1–a model zoo for robust monocular relative depth estimation. arXiv preprint arXiv:2307.14460 (2023)
2. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
4. Face, H.: Compvis/stable-diffusion-v1-4 (2023), `https://huggingface.co/CompVis/stable-diffusion-v1-4`, retrieved September 15, 2023

5. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
6. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
7. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
8. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
9. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
10. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision (ECCV) (2022)
11. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
12. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision (ECCV) (2020)
13. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
14. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
15. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML) (2021)
16. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
17. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (ICLR) (2021)
18. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: International Conference on Machine Learning (ICML) (2023)