SUPPLEMENTARY MATERIAL FOR GANZZLE: REFRAMING JIGSAW PUZZLE SOLVING AS A RETRIEVAL TASK USING A GENERATIVE MENTAL IMAGE

Davide Talon[†,‡], *Alessio Del Bue*[†], *Stuart James*[†]

[†]Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT), Genova, Italy
[‡]Università degli studi di Genova, Italy

1. INTRODUCTION

In this document, we provide extended details on the dataset and evaluation. In section 2 we detail quantitative statistics of the two proposed datasets PuzzleCelebA and PuzzleWikiArts. In section 3, we define the implementation details of baselines and an analysis of their performance through the two evaluation metrics, Direct Accuracy (Tables 4, 2) and Neighbor accuracy (Tables 5, 3) as well as the performance on missing, noisy or eroded pieces (Table 6). We additionally include analysis on puzzle sizes 2 and 4 not shown in the primary text. We also provide additional qualitative analysis over multiple puzzles for both the PuzzleCelebA and PuzzleWikiArts datasets in Figure 2 and Figure 3.

2. PUZZLE DATASETS

We propose two datasets providing different challenges. Firstly, a visually simple (and consistent) dataset based on CelebA [1], which has been demonstrated that generative methods are able to synthesize with high accuracy. Secondly, an arts-centered dataset based on WikiArts [2], this provides a challenging environment for generalization of methods across different styles and content. In contrast to prior approaches both of these datasets are large (thousands of images), therefore providing a testing ground for both puzzle and spatial reasoning problems.[1] For each dataset we provide a text file split for training and test. To enable fair comparison between different approaches, for test samples we provide fixed shuffling permutations.

2.1. PuzzleCelebA

To evaluate our model, we relied on a high quality version of CelebA [3], a large-scale face image dataset. Counter intuitively, faces represent a challenging testbed for jigsaw puzzle solving. Two sources of ambiguities are concurrently involved: faces are highly symmetrical and profile pictures are characterized by blurred (or plain) background. We noticed the consistent structure of images to ease the generation of guiding images to match against. From the $30k$ images we applied a random $80-20\%$ train-test split and for each image we generate six grid sizes, i.e., complexities, $[2, 4, 6, 8, 10, 12]$ where each generated puzzle is randomly shuffled. We provide the $35,994$ test puzzle permutations for comparative evaluation.

2.2. PuzzleWikiArts

We build from a subset of the WikiArts dataset [2], proposing a split and different grid sizes for evaluation. We select art style categories that avoid highly complex symmetries that make placing pieces ambiguous, therefore the dataset contains varying difficult examples including more unique humanoid structures as well as patterns that will challenge puzzle solving algorithms with near duplicate pieces (see Fig. **??**. We take the $63,126$ images of the dataset and split them randomly into an $80-20\%$ train-test split resulting in $50,502$ training images. Similarly to CelebA jigsaw data, for each image we generated three grid sizes. A total of $37,875$ test puzzles is present.

3. BASELINE DETAILS

We compared our proposed with representative methods of different approaches. To this end, we considered three different optimization-based algorithms: **Gallagher** [4], **Pomeranz and Tal** [5] and **Paikin et al.** [6]. For deep learning solutions, we compare to Permutation-Optimization with Linear Assignment (**PO-LA**) [7] from set representation. Furthermore, we considered a deep learning baseline based on Hungarian attention [8], here denoted as **hung-perm**.

We built on the original implementations of chosen optimization approaches: Paikin and Tal [6], Pomeranz and Tal [5] and Gallagher [4]. Hyper-parameters are set to those of the original proposals. Test jigsaw with square 32 pixels patches are fed to solving algorithms, Predicted reordering permutations are hence retrieved. Due to limited and controlled evaluation protocols of classic methods, we found optimization

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870723.

[1]Datasets will be released upon acceptance

(a) (b)

Fig. 1: Example puzzles from (a) PuzzleCelebA and (b) PuzzleWikiArts

approaches to fail with black tiles that could randomly be present in images. Indeed, original implementations do not characterize missing pieces. Therefore, a major adaptation was needed to set apart black pieces from missing ones. To this end, we set the color of the patch middle point to white. Unluckly, we could not evaluate such approaches for low-size puzzles, due to instability of implementations.

We adapted PO-LA [7] algorithm to handle the large cardinality of puzzles pieces. Specifically, owing to the algorithm's memory requirements, we rescaled tiles to 16×16 and set the hidden size of the network producing the ordering cost to 64. Adam [9] with a fixed learning rate of $1e-3$ is employed for 20 epochs using a batch size of 32. The PO-LA approach was not assessed for larger sizes due to the computational limitations of its memory footprint.

In hung-perm, a ResNet-50 backbone encodes tiles independently. For each tile, the 2048-dimensional embedding is hence mapped to a a vector of size 256 with a fully connected linear layer. Finally, after gathering of all embeddings by concatenation, the model outputs the 1-1 assignment problem to optimize for ,i.e., the assignment cost matrix. Similary to GANzzle, the predicted permuation is obtained as hungarization of the associated doubly stochastic matrix. The model is trained end-to-end via Hungarian attention for 150 epochs. Adam with learning rate of 0.01 and batch size 64 is employed. In deep learning solutions, we trained a model for each size.

Model	**Average Sample (ms)**
Paikin and Tal [6]	27.47 ± 7.70
Pomeranz et al. [5]	221.64 ± 300.79
Gallagher [4]	235.19 ± 358.72
PO-LA [7]	22.38 ± 8.08
Hung-Perm	9.97 ± 1.38
Ganzzle	25.16 ± 1.1

Table 1: Compuation time requirements in ms for the different approaches.

4. IMPLEMENTATION DETAILS

For both baseline and GANzzle experiments we conduct training on 20 core Intel Xeon, 394Gb RAM and 4x NVIDIA Tesla V100 16Gb. We pre-train the GANzzle GAN for 144hrs (72hrs with out discriminator and 72hrs with), we note that WikiCelebA converges significantly faster and therefore simpler datsets than PuzzleWikiArts could be trained with a much shorter time. We then train the full GANzzle model on WikiCelebA for 24hrs, and 72hrs on PuzzleWikiArts. We use the Adam optimizer with a learning rate of $1e-3$ for the Generator (G) and $4e-3$ for the Discriminator (D), in joint training we use $1e-3$ for the learning rate.

5. COMPUTATIONAL TIME

We compare the computation wall time for averaged over 24 samples in Table 1. It can be seen against [5] and [4] the deep learning methods (including GANzzle) have a significant time cost. In contrast to Paikin and Tal the performance is comparible, however, it should be noted that as the puzzle size increases the time increases in contrast. Deep learning methods are largely similar, with the minimal (without GAN) Hung-Perm taking half the computational time.

Model	2x2	4x4	6x6	8x8	10x10	12x12
Paikin and Tal [6]	-	-	98.03	97.35	95.31	90.52
Pomeranz et al. [5]	-	83.34	79.23	72.64	67.70	62.13
Gallagher [4]	98.10	93.94	89.47	83.22	78.25	73.40
Hung-perm	47.09	30.37	8.42	3.22	1.90	1.25
PO-LA [7]	85.67	34.58	12.19	5.77	3.28	-
GANzzle-Single (Ours)	-	25.34	11.78	6.23	8.97	-
GANzzle (Ours)	76.86	29.72	13.48	6.93	4.10	2.58

Table 2: Direct accuracy comparison on PuzzleWikiArts.

Model	2x2	4x4	6x6	8x8	10x10	12x12
Paikin and Tal [6]	-	-	99.37	99.09	98.23	95.97
Pomeranz et al. [5]	-	94.83	93.39	89.96	87.25	84.07
Hung-perm	45.36	18.72	4.25	1.97	1.43	0.90
PO-LA [7]	82.32	26.40	7.94	4.01	2.58	-
GANzzle-Single (Ours)	-	17.96	9.53	6.30	8.21	-
GANzzle (Ours)	72.33	22.61	11.08	7.10	5.32	4.18

Table 3: Neighbor accuracy comparison on PuzzleWikiArts.

Model	2x2	4x4	6x6	8x8	10x10	12x12
Paikin and Tal [6]	-	-	99.12	98.67	98.39	96.51
Pomeranz et al. [5]	-	84.67	84.59	79.43	74.80	66.43
Gallagher [4]	99.53	95.76	90.08	77.69	66.68	53.53
Hung-perm	99.95	88.05	33.11	12.89	4.14	2.18
PO-LA [7]	99.80	92.20	71.96	50.12	38.05	-
GANzzle-Single (Ours)	-	91.05	71.00	51.81	43.74	-
GANzzle (Ours)	99.67	87.80	72.18	53.26	32.84	12.94

Table 4: Direct accuracy comparison on PuzzleCelebA.

Model	2x2	4x4	6x6	8x8	10x10	12x12
Paikin and Tal [6]	-	-	99.70	99.38	99.15	96.51
Pomeranz et al. [5]	-	96.39	96.31	93.87	91.38	87.79
Hung-perm	99.92	84.57	22.35	7.49	2.33	0.95
PO-LA [7]	99.71	89.82	66.43	44.02	32.72	-
GANzzle-Single (Ours)	-	88.13	64.70	44.15	37.51	-
GANzzle (Ours)	99.52	83.87	66.04	46.20	26.46	9.93

Table 5: Neighbor accuracy comparison on PuzzleCelebA.

Model	Missing (%)			Noisy (σ)			Eroded (px)		
	10%	20%	30%	0.05	0.1	0.2	1	2	5
Paikin and Tal [6]	-	-	-	47.58	9.83	3.36	2.81	2.78	2.79
Pomeranz et al. [5]	43.12	21.44	17.66	84.27	87.62	90.49	3.18	3.18	5.64
Gallagher [4]	80.94	66.76	52.20	95.91	96.28	96.30	31.62	23.48	7.58
Hung-perm	7.85	7.12	6.55	6.63	5.35	4.24	3.69	4.03	3.17
PO-LA [7]	11.42	13.45	16.64	11.77	10.75	8.43	4.96	3.23	2.58
GANzzle-Single (Ours)	10.20	8.34	7.21	10.79	9.63	7.90	4.73	4.52	3.12
GANzzle (Ours)	11.60	9.44	7.88	11.80	10.22	8.19	5.07	4.99	3.29

Table 6: PuzzleWikiArts dataset. Comparison of different noise techniques for pieces using direct accuracy comparison for missing pieces, additive Gaussian noise and eroded pieces on a 6×6 puzzle.

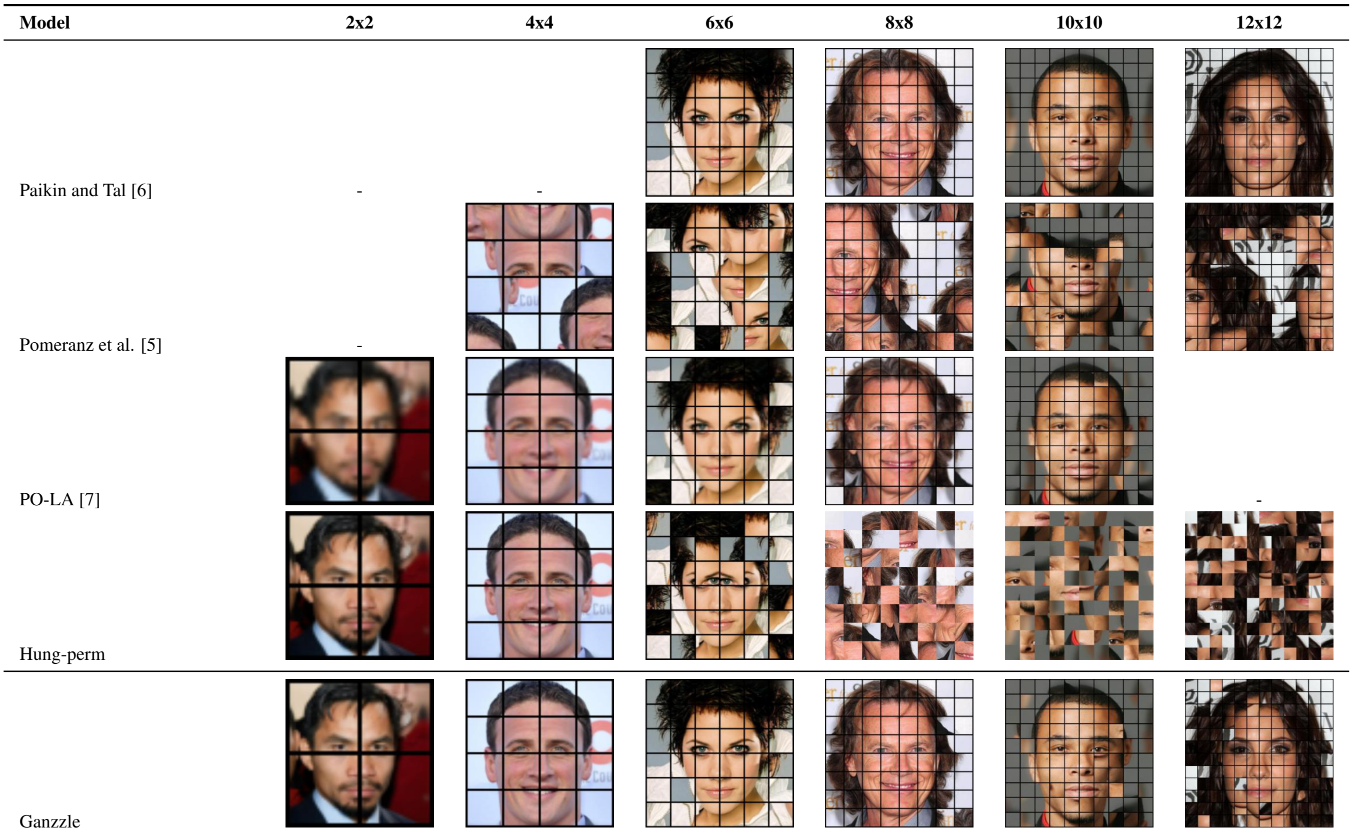

Fig. 2: Qualitative examples for PuzzleCelebA

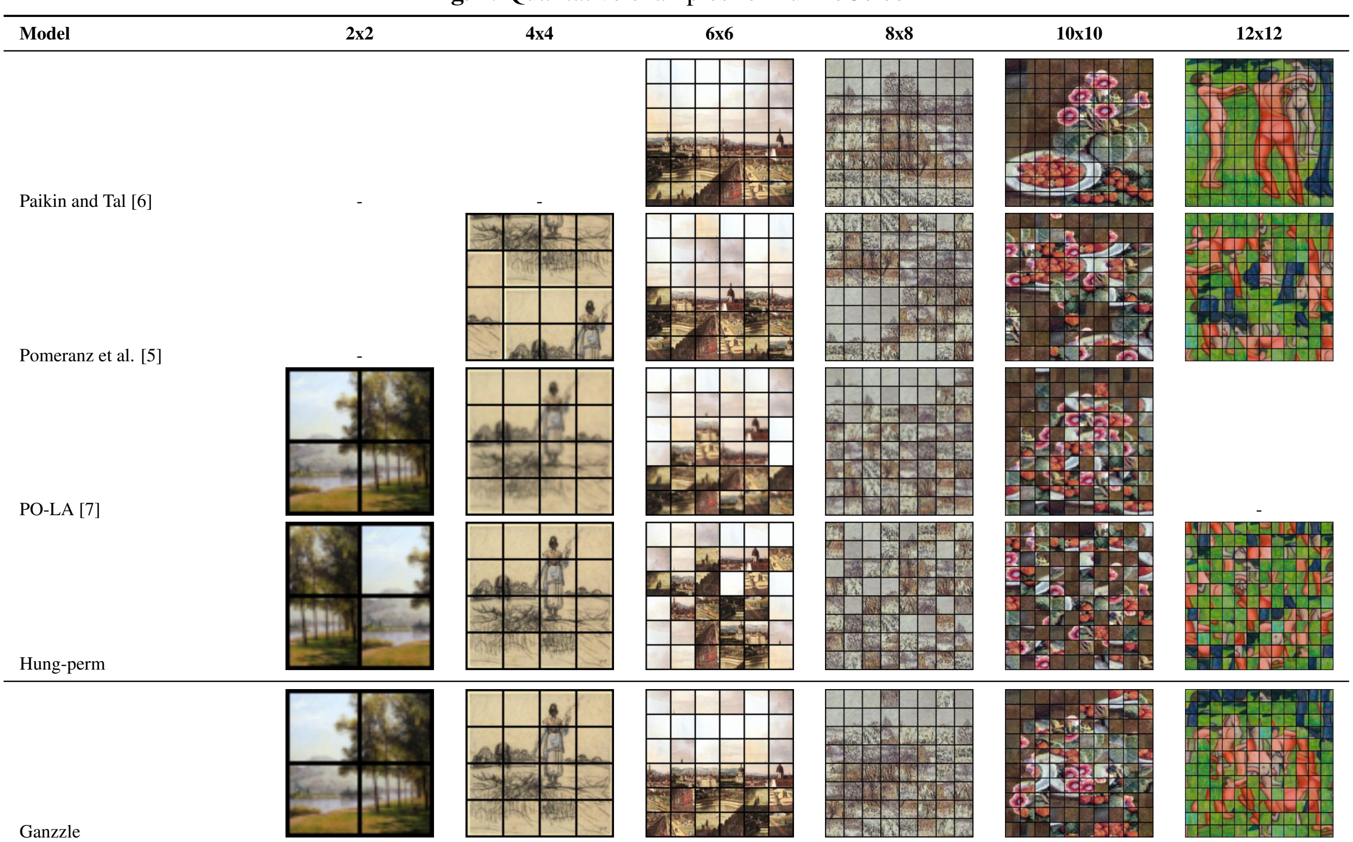

Fig. 3: Qualitative examples for PuzzleWikiArts

6. REFERENCES

[1] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[2] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, "Improved artgan for conditional synthesis of natural image and artwork," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 394–409, Jan 2019.

[3] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] A. C. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 382–389.

[5] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *CVPR 2011*, June 2011, pp. 9–16.

[6] G. Paikin and A. Tal, "Solving multiple square jigsaw puzzles with missing pieces," *IEEE CVPR*, pp. 4832–4839, 2015.

[7] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Learning representations of sets through optimized permutations," in *ICLR*, 2019. [Online]. Available: https://openreview.net/forum?id=HJMCcjAcYX

[8] T. Yu, R. Wang, J. Yan, and B. Li, "Learning deep graph matching with channel-independent embedding and hungarian attention," in *International conference on learning representations*, 2019.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.