Pattern Recognition Letters
journal homepage: www.elsevier.com

GANzzle++: Generative approaches for jigsaw puzzle solving as local to global assignment in latent spatial representations

Davide Talon[a,b,**], Alessio Del Bue[a], Stuart James[a,c]

[a] *Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT), Genova, Italy*
[b] *Universita' degli Studi di Genova (UniGe), Genova, Italy*
[c] *Department of Computer Science, Durham University, United Kingdom*

ABSTRACT

Jigsaw puzzles are a popular and enjoyable pastime that humans can easily solve, even with many pieces. However, solving a jigsaw is a combinatorial problem, and the space of possible solutions is exponential in the number of pieces, intractable for pairwise solutions. In contrast to the classical pairwise local matching of pieces based on edge heuristics, we estimate an approximate solution image, i.e., a *mental image*, of the puzzle and exploit it to guide the placement of pieces as a piece–to-global assignment problem. Therefore, from unordered pieces, we consider conditioned generation approaches, including Generative Adversarial Networks (GAN) models, Slot Attention (SA) and Vision Transformers (ViT), to recover the solution image. Given the generated solution representation, we cast the jigsaw solving as a 1-to-1 assignment matching problem using Hungarian attention, which places pieces in corresponding positions in the global solution estimate. Results show that the newly proposed GANzzle-SA and GANzzle-VIT benefit from the early fusion strategy where pieces are jointly compressed and gathered for global structure recovery. A single deep learning model generalizes to puzzles of different sizes and improves the performances by a large margin. Evaluated on PuzzleCelebA and PuzzleWikiArts, our approaches bridge the gap of deep learning strategies with respect to optimization-based classic puzzle solvers.

the differences between local and global

the details of attention

how to combinate these three methods

what is the specifical scale of the dataset?

1. Introduction

As a general problem of the *sorting* primitive, solving jigsaw puzzles have implications for a wide range of real-life applications. While initial effort on the topic has been motivated by the necessity of automatic assembly strategies for fragmented historical artifacts, several applications have been framed in the puzzle-solving setting, namely assembling broken objects [12, 24], biology tasks [20, 30], shredded documents recovery [8], and image and fresco reconstruction [10, 31], speech [35] and image editing [5]. While the geometric nature, e.g., the shape, of many of these problems can be beneficial, the solving of relatively simple square puzzles relying solely on the visual appearance information remains an open challenge.

why does this method apply in other areas? similarities?

why are visual representations problems? space?memory?detection?

**Corresponding author
e-mail: talon.davide@gmail.com (Davide Talon), alessio.delbue@iit.it (Alessio Del Bue), stuart.a.james@durham.ac.uk (Stuart James)

the relationship of the number vs the difficulities

Though jigsaw puzzles were conceived as a children's game to develop spatial reasoning skills, the combinatorial complexity challenges automatic solvers as the number of permutations grows exponentially with the number of pieces. Previous works mostly explored heuristics on how parts should match as a local contrasting rule to place them [14, 26]. To this end, practitioners have primarily focused on the layout of pieces or the color continuity of edges across adjacent pieces. Indeed, such algorithms consider pairwise relationships between pieces and do not account for their global arrangement, i.e., placement of patches is a sequential procedure solving for the remaining parts. While achieving impressive results with a large number of pieces, two main drawbacks affect classic approaches. On one side, to avoid suboptimal solutions due to the sequential placement (sensitivity to the seed), pieces are re-placed multiple times, leading to infeasible time demand. On the other, the exploitation of edge continuity is brittle when no such information is provided, as in cultural heritage scenarios [13], e.g.,

what are the local issues?

time costly for repeating

the missing information of the edges

historical settings where pieces have undergone erosion.

Our work is inspired by GANzzle [28] that reframes puzzle solving as a one-to-one assignment using a deep learning strategy. Talon et al. [28] predicts the arrangement of pieces based on a global estimate of the final solution image, i.e., the global information, and assigns pieces according to the corresponding position in the global representation. Noticeably, the placement of pieces is performed globally, accounting for all pieces together, and is not time-demanding as it requires a network forward pass only. Furthermore, the model does not explicitly build on the boundary information of patches.

the details of global approach

We extend Talon et al. [28]'s work with a study on the generative module of the approach that crucially estimates the global solution of the jigsaw for later matching. We present two different variants of the generative module leveraging Slot Attention [19] and Vision Transformers (ViT) [11], respectively. Contrary to the two-step processing of GANzzle, where patches are first embedded in the latent space and then pooled to a fixed-size representation, the new modules jointly consider all pieces to recover the global compressed representation. The proposed improved generative modules manage a variable number of pieces while avoiding the critical pruning of information given by pooling to a low-dimensional fixed-sized embedding representation. Starting from unordered pieces, the generative modules output a global representation with lower compression loss, maintaining necessary information for subsequent matching of the pieces. GANzzle-SA and GANzzle-VIT improve over the vanilla version of GANzzle and compare favorably to other deep learning approaches on PuzzleCelebA and PuzzleWikiArts [28], on both direct and neighbour accuracy. We further show that GANzzle-VIT bridges the performance gap between deep learning and optimization-based puzzle solvers.

details of variants

how to compare

how to predict the neighbours

The contribution of this work is three-fold: i) We present two novel generative approaches for estimating the global solution of the jigsaw starting from an unordered set of pieces. ii) We study the effect of different generative approaches for local-to-global jigsaw matching, and iii) We evaluate the benefit of the proposed methods, allowing bridging the gap of deep learning strategies to optimization-based puzzle solvers.

The rest of this paper is organized as follows: Section 2 provides an overview of recent literature on jigsaw puzzle solving, Section 3 presents the proposed generative strategies and 4 evaluates the effectiveness of the approaches. Finally, Section 5 concludes the work.

how to evaluate

what is differences between the optimization-based and deep learning

2. Related Work

Various solutions have been proposed for the visual jigsaw problem in recent years. Two prominent families of approaches are present in the literature: optimization-based solutions and deep learning strategies. The former builds on heuristics on piece matching and casts the jigsaw as an optimization problem to optimize for, and the latter leverages neural network feature extraction to find the correct permutation of pieces.

Optimization-based. Cho et al. [6] formulate puzzle-solving as a graphical model labeling problem. Pieces are labels to assign to graph nodes representing slots. On one side, belief propagation allows sharing neighbor information to already allocated pieces and optimizing for the solution; on the other, a dissimilarity-based compatibility metric evaluates the fit of adjacent pieces. Building on the pair matching metric, Pomeranz et al. [26] propose a prediction-based heuristic shifting from color differences to a compatibility metric evaluating whether pixels in the boundary of a piece correctly predict the edge pixels of an adjacent piece. The work introduces the concept of best buddies, i.e., a pair of pieces that agree on being the most likely neighbors in a spatial relationship. An iterative placement strategy refines placement of pieces to tackle the seed sensitivity of a greedy solution. Gallagher [14] considers a pairing heuristic assessing the continuity of gradients in adjacent pieces. Components are penalized based on the Mahalanobis distance. Therefore, solving is cast as a minimum spanning tree problem where edges represent spatial relationships between pieces: starting from trivial forests, minimum weight edges leading to admissible merges are selected. Trimming and filling adapt the assembled puzzle to the target frame and fill missing slots. A placement policy based on uncertainty tied with a principled choice of the seed has proved effective for solving puzzles with missing pieces [23]. The first-placed piece should be distinctive and in a distinctive area, i.e., a piece surrounded by its best buddies that, in turn, have all their best buddies. Hence, a greedy solver iteratively places the most reliable piece, i.e., the one minimizing the likelihood of misplacement with respect to other set pieces.

what is the seed

clusters?

In contrast, this work builds on a deep learning global solution strategy that does not explicitly build on edge information to perform local matching. Differently from other global approaches accounting for local constraints of adjacent pieces, we optimize for an unconstrained 1-1 assignment based on the feature similarity of pieces and slots from the estimated global solution. Contrary to the demanding computational times of optimization-based strategies, a deep neural network forward pass allows for a time-efficient estimation of the global solution and recovers the permutation of pieces using a relaxed version of the Hungarian [16] algorithm, that is cubic in the number of pieces.

deep learning saves time?

Deep Learning. Building on previous approaches, Zhang et al. [34] consider a learnable cost function. The approach jointly optimizes both the cost matrix assessing pairwise relationships between objects and the correct permutation in a bi-level optimization scheme. Rafique et al. [27] consider a GAN setting where the generator outputs a n-dimensional placement vector, i.e., a vector whose i-th element indicates the index of the piece to associate, and the discriminator takes apart real-placement outputs from non-admissible ones. Learning the complete piece-to-location mapping task is challenging and unstable as the permutation space increases and is additionally confounded by the inability to regress a vector of integers for neural networks. In contrast, we argue that the synthesized image is a better solution for solving the locations of the pieces and shows how this approach can scale to more complex puzzles. Alternatively, Bridger et al. [1] tries to infill using a GAN between pairs of pieces for solving the assembly problem. However, the placement considers pairs sequentially and

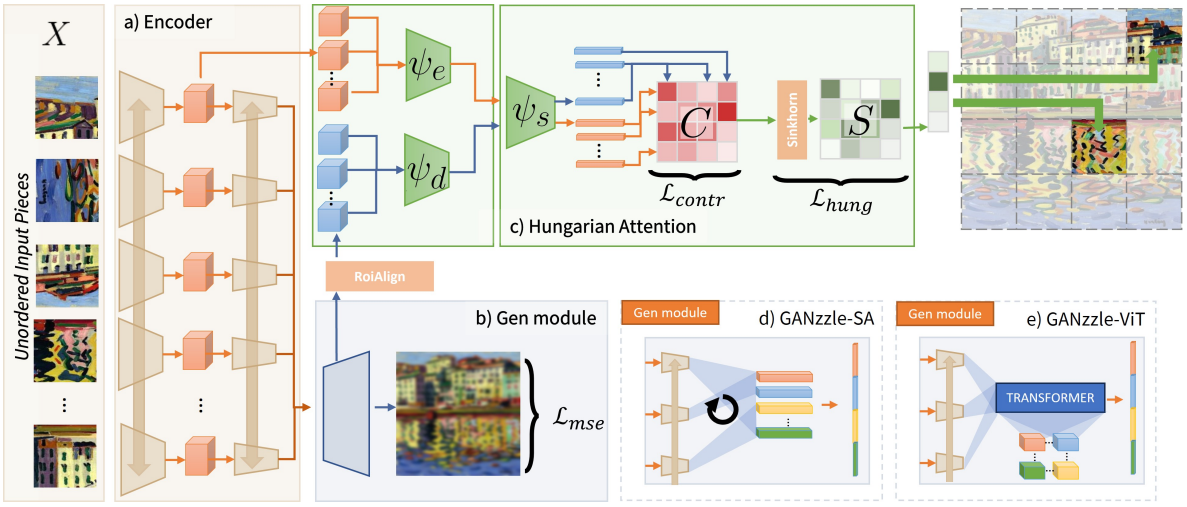

Fig. 1. GANzzle solves a jigsaw by first estimating the global solution and later piece-to-global matching. (a) A shared encoder embedds pieces X. (b) A generative module gathers information from all pieces to estimate the target solution. An intermediate feature representation of the generative model is cropped using RoiAlign to act as targets slots. ψ_d, ψ_e and ψ_s map pieces and slots to a joint space. (c) A cost matrix C models the cost of assignment and Hungarian Attention solves for the final permutation by relaxing the problem to a doubly stochastic matrix S via Sinkhorn normalization. The choice of the generative module is crucial for placement: (d) based Slot Attention (SA) and (e) Vision Transformer (ViT).

has demanding time requirements for computation. Recently, jigsaw puzzles have been considered as an unsupervised representation learning method to get semantic features for downstream tasks [2, 3, 9, 22]. The relative spatial position of pairs of adjacent pieces helps understand the composition of an image in its objects and learn their geometry [9]. However, these approaches do not enforce the assignment constraint, i.e., one piece only for a slot, making the strategy unsuitable for handling conflicting assignment [2, 3, 9] or do not handle a variable number of pieces [3, 22]. unsupervised - similar pieces crash in same place

Cruz et al. [7] propose to optimize over the continuous surrogate of discrete permutation matrices: doubly stochastic matrices that are non-negative real-valued matrices whose rows and columns sum to one. Such matrices could be obtained through the differentiable Sinkhorn normalization procedure. The closest permutation matrix is selected via linear integer programming at inference time. Similarly, Mena et al. [21] parameterize the permutation matrix to make it amenable for the reparameterization trick. In Deepzzle [25], a Siamese network predicts the relative placement of pieces with respect to a central anchor. Therefore, the algorithm builds an assembly tree where pieces are related to positions in the solution and edges model placement cost. Then, the tree is optimized for the shortest path. In Li et al. [17] a jigsaws classification branch classifies which permutation has been applied to pieces. Consequently, a flow-based warp is applied to features to recover the original image. Hence, sorted features condition the generator of the GAN to produce realistic-looking images. Starting from the unordered pieces, GANzzle [28] estimates the jigsaw global solution using a GAN. In a later phase, the approach cast placement as a 1-to-1 assignment where pieces are matched to slots, i.e., holes in the global estimated solution, based on their similarity. In contrast to the symbolic representation of patches in [27] GANzzle leverages a GAN to synthesize the solution global image and does not limit to inpaint eroded parts [1] for a slow local matching strategy. The estimated solution accounts for global information in a single inference step. While in Li et al. [17] the adversarial branch aids the classification of the given permutation by projecting it in the image space, in GANzzle, the generator is learning to permute pieces correctly. As a benefit, the solution can cope with an arbitrary permutation of pieces.

what is flow-based warp

solve the order issues! but time?numbers?accuracy?

In contrast, this work builds on GANzzle and investigates the role of the generative module in estimating the global solution. Contrary to the GAN generator, we present two different generative approaches for inferring the global estimate from unordered pieces based on Slot Attention [19] and Vision Transformer [11] and show that accurate global estimation is critical for the final puzzle solution. The improvement closes the gap with respect to optimization-based solutions, especially with a larger number of pieces.

specifical scale?

3. Method

The GANzzle framework predicts the permutation of a set of n image patches $X = \{X_1, \ldots, X_n\}$, $X_i \in \mathbb{R}^{h \times w}$, $i = 1, \ldots, n$ to recover the original image they are part of $Y \in \mathbb{R}^{th \times tw}$, with $n = t^2$ being the number of pieces and h, w patches height and width respectively. The model learns to predict piece locations supported by an estimate of the target image, see fig. 1 for a visualization. Starting from unsorted pieces, the approach independently encodes each piece X_i through a shared encoder (fig. 1a). A generative module then gathers pieces information and estimates the global jigsaw target image (fig. 1b). The generative module design choice is critical (fig. 1d, 1e and sec. 3.1). In a second phase, the model learns to match the pieces to targets within the global encoding. To this end, a cost matrix between patches and target holes accounts for the cost of their assignment and a differentiable version of the Hungarian algorithm solves for the optimal placement of pieces by attending only relevant assignment information (fig. 1c, sec. 3.2).

prediction of target -> location match?
how does it work ? advantages or disadvantages?

3.1. Estimate the global target information

The generative model estimates the global solution of the jigsaw starting from unordered patches. With the aim to handle a variable number of pieces and seek for permutation invariance with respect to the pieces order, the module maps the input patches to a lower dimensional feature space according to an encoder with shared weights $E(\cdot)$, $z_i = E(X_i)$ where $z_i \in \mathbb{R}^{d_h}$ is the d_h-dimensional feature representation of the i-th piece. Hence, to solve for positioning, the module accounts for the global information of all pieces and projects the representation to a unique global representation. Contrary to prior work [28], we experiment with various generative approaches that differ in the gathering of information and estimation of the target global solution:

Generative Adversarial Network(GANzzle). Piece embeddings are first pooled to a fixed-sized representation and then a convolutional Generative Adversarial Network (GAN) approximates the solution of the Jigsaw. Specifically, we follow Talon et al. [28] that gathers information from all pieces by performing an average operation component-wise on all pieces, $z = avg(z_1, \dots, z_n)$. Hence, a generator synthetizes the global solution as $\hat{Y} = G(z)$. We consider a MSG-GAN style of approach to guide the generation at multiple resolutions. The generator $G(\cdot)$ and the discriminator $D(\cdot)$ are trained in the standard min-max fashion. In contrast to the vanilla GAN, multiple scales are considered by extending equations to different granularities:

$$\mathcal{L}_{gen} = \min_G \max_D \sum_{l=1}^{L} L_{gan}(G^l, D^l) + \lambda_p \mathcal{L}_{mse}(G^l),$$

where G^l is the RGB-converted intermediate representation of the generator at layer l (depth L) and D^l the corresponding discriminator. A pixel-wise mean squared error term $\mathcal{L}_{mse}(\cdot)$ is added, weighted by λ_p.

Slot Attention decoder (GANzzle-SA). Slot attention allows to overcome the bottleneck compression due to a pooling strategy by building on an iterative attention-based specialization scheme. The set of n pieces embedding are mapped to K D_s-dimensional vectors, where K is the number of slots and D_s is the slot dimension. At first, slots are initialized based on a SinCos positional embedding [32] $slot_k^0 = SinCos(k), k = 1, \dots, K$ and the pieces exchange global information thanks to a 2-layers transformer encoder $z = \text{Transformer}(z_1, \dots, z_n)$ Hence, an iterative procedure clusters the pieces based on the dot-scaled product attention. At each step $s = 1, \dots, S$:

$$\texttt{attn}_{ij} = \text{Softmax}(\frac{1}{\sqrt{D}} k(\texttt{z}) q(\texttt{slots})^T), \quad (1)$$

where $q(\texttt{slots})$ and $k(\texttt{z})$ represent respectively the slot queries and pieces keys, and D is the key dimension. Update weights are computed as:

$$W_{ij} = \frac{\texttt{attn}_{ij}}{\sum_{r=1}^{n} \texttt{attn}_{rj}}. \quad (2)$$

Hence, $\texttt{update} = W^T v(\texttt{z})$ are used to update the slots according to a linear projection $v(\texttt{z})$ of attended pieces z. Slots are implemented as a D_s-dimensional Gated Recurrent Unit [4] Cells: $\texttt{slots}^s = \text{GRU}(\texttt{slots}^{s-1}, \texttt{update})$ The K D_s-dimensional slots are hence concatenated, reshaped to spatial dimension and decoded to the size of the ground truth image to recover. The model is trained with the standard MSE reconstruction loss $\mathcal{L}_{gen} = \text{MSE}(d(\texttt{slots}^S), Y)$, where $d(\cdot)$ decodes the slots to the image space.

Vision Transformer decoder (GANzzle-VIT). We build on a pretrained Vision Transformer encoder model [11] that leverages the patch-oriented processing and an early-fusion strategy. A single convolutional layer tokenizes the input patches as $z_i = PE(X_i), i \in 1, \dots, n$. Hence, a transformer pools the information of all pieces. Let $\texttt{POOL}$ denote a set of K learnable pooling tokens, the transformer allows for information exchange across the pieces:

$$\texttt{POS-POOL} = \text{positional-embedding}(\texttt{POOL}) \quad (3)$$

$$z, z_1, \dots, z_n = \text{transformer}(\texttt{POS-POOL}, z_1, \dots, z_n) \quad (4)$$

where positional-embedding$(\cdot)$ denotes the 2-dimensional positional embedding [15]. Hence, the global representation z is reshaped to a spatial dimension and decoded via $d(\cdot)$ for reconstruction, $\hat{Y} = d(z)$. Contrary to GANzzle, the model does not employ an adversarial approach, and is trained by minimizing the mean squared error loss $\mathcal{L}_{gen} = \text{MSE}(\hat{Y}, Y)$.

Crucially, both GANzzle-SA and GANzzle-VIT do not employ the two-steps processing of patches of GANzzle where patches are first embedded in the latent space and then pooled to a fixed-size representation. As gathering the information from all pieces requires a high compression rate of their informative content, GANzzle-SA and GANzzle-VIT jointly consider all pieces to recover the global representation. On a computational side, GANzzle's matching remains the most demanding operation of the approach as during training the optimal assignment should be computed. For the generative module, GANzzle-SA K slots attend the n pieces and is linear in the number of pieces but suffer from the sequential iterative update of slots. On the contrary, the highly parallelizable GANzzle-VIT has a memory footprint quadratic in the number of pieces.

like hash map clash

how to low the computing space for global features? in addition to compress method? any math models?

3.2. Piece assignment

We cast the problem as 1-to-1 mapping by constructing an assignment cost matrix C based on the similarity between pieces and placement positions given by the RoiAlign chunks of the target estimate. To this end, the intermediate representation of pieces and target positions are embedded with shallow networks ψ_e and ψ_d for piece and target positions, respectively. Finally, a common converting module ψ_s guarantees the alignment of the embedding spaces. Hence, the similarity matrix is computed as dot product of all possible piece-slot pairs at runtime, making it dynamic to the size of the puzzle. A contrastive loss regularizes the feature space so as to enforce similar embeddings for piece-slot correct pairs while increasing the distance between non-corresponding pairs:

$$\mathcal{L}_{contr} = -\mathbb{E}_i \left[\log \frac{exp\left(\psi_s^i \cdot \psi_s^j / \tau\right)}{exp(\psi_s^i \cdot \psi_s^j / \tau) + \sum_{k \neq j} exp(\psi_s^i \cdot \psi_s^k / \tau)} \right] \quad (5)$$

with ψ_s^i and ψ_s^j embeddings of considered piece i and its corresponding slot j, τ the temperature parameter and $\mathbb{E}_i$ the mean over puzzle pieces.

Assignments based on the cost matrix could then be efficiently computed by employing the Hungarian algorithm [16]. However, the approach is non-differentiable due to the discrete nature of assignments. We, employ Hungarian Attention [33] to learn the assignment task in a supervised way. Hence, the problem is continuously relaxed. A doubly stochastic matrix is obtained via the iterative Sinkhorn normalization:

$$S^0(C) = \exp(C) \tag{6}$$

$$S^l(C) = F_c\left(F_r\left(S^{l-1}\right)\right) \tag{7}$$

$$\mathbf{S} = \lim_{l\to\infty} S^l(C), \tag{8}$$

where F_c and F_r are the row and column-wise normalization $F_c(C) = C \oslash \left(\mathbf{1}_N \mathbf{1}_N^T C\right)$ and $F_r(C) = C \oslash \left(C \mathbf{1}_N \mathbf{1}_N^T\right)$ respectively, with $\oslash$ denoting the element-wise division and 1_N the n-dimensional unit column vector. To avoid overconfidence due to the sparsity of the permutation matrix, a hard attention mask is generated by comparing the predicted permutation Hung $(\mathbf{S})$ computed by applying the Hungarian algorithm to $\mathbf{S}$, to the ground-truth assignment $\mathbf{S}^G$ through an element-wise logic-OR operator:

$$\mathbf{Z} = OR\left(\text{Hung}\left(\mathbf{S}\right), \mathbf{S}^G\right). \tag{9}$$

Notice that the hard mask focuses on most relevant elements in the matrix, i.e., both correct and misplaced pieces are modeled. Hence, a binary cross-entropy loss with respect to the ground-truth assignment matrix is attended through the mask:

$$\mathcal{L}_{hung} = \sum_{i,j\in[n]} \mathbf{Z}_{ij}\left(\mathbf{S}_{ij}^G \log \mathbf{S}_{ij} + \left(1 - \mathbf{S}_{ij}^G\right) \log\left(1 - \mathbf{S}_{ij}\right)\right), \tag{10}$$

where $[n]$ is the set of indexes from 1 to n.

By optimizing the above permutation loss, our model learns to correctly match the Hungarian's assignment computed from S to the ground truth permutation. At inference time, the estimated assignment is hence the Hungarian binarization of the doubly stochastic matrix Hung $(\mathbf{S})$.

The complete loss for the GANzzle model is therefore:

$$\mathcal{L} = \mathcal{L}_{gen} + \mathcal{L}_{hung} + \mathcal{L}_{contr}. \tag{11}$$

We found all the losses to be necessary for an end-to-end approach for puzzle solving. In particular, $\mathcal{L}_{gen}$ guides the generative module to synthethize the estimated solution, $\mathcal{L}_{hung}$ provides training signal for the global-to-local assignment and $\mathcal{L}_{contr}$ regularizes the embeddings to be discriminative. Critically, we train the model on puzzles of various sizes, making the approach size-agnostic. We leverage a gradient accumulation strategy where samples are grouped into batches based on the jigsaw complexity. For each batch, we evaluate the loss and perform a backward pass of gradients. However, weights are updated only once all sizes have been considered.

4. Experiments

We assess the benefit of improving the global estimation for placement in GANzzle-SA and GANzzle-VIT on PuzzleCelebA and PuzzleWikiArts datasets [28] in terms of quantitative metrics and qualitative results.

Datasets. We consider PuzzleCelebA and PuzzleWikiArts. The former is a visually simple (and consistent) face image dataset based on CelebA [18]. While easy on a generative side, two sources of ambiguities are concurrently involved for puzzle solving: faces are highly symmetrical and profile pictures are characterized by blurred (or plain) background that makes patches ambiguous. The latter is an arts-centered dataset based on WikiArts [29], that provides a challenging environment for generalization of methods across different styles and content. PuzzleWikiArts is characterized by its high variability as it contains varying difficult examples including more unique humanoid structures as well as patterns that will challenge puzzle solving algorithms with near duplicate pieces. dataset size?

Evaluation metrics. We use the standard direct comparison metric [6] where an assignment is considered correct if it is placed in the correct absolute position. We further include results on the neighbor accuracy metric evaluating the average fraction of neighbor pieces that are correctly placed: two patches are correct neighbors if and only if the two pieces are in the same relative position in the ground truth and the estimated solution. how to carry out?

Baselines. We compare against optimization methods [14, 23, 26] and deep learning strategies such as Zhang et al. [34], Hung-perm [28] and GANzzle[28]. We directly report baselines results from Talon et al. [28] where Hung-perm and Zhang et al. [34] are size-specific with the latter limited on 12×12 due to memory explosion.

comparison based on same enviroment?
details about performance and confition?

4.1. Results

Table 1 shows the direct comparison accuracy for both PuzzleCelebA and PuzzleWikiArts. The GANzzle strategy generalizes across sizes and it is competitive with other deep learning solutions. As can be noted from the large margin improvement of GANzzle with respect to Hung-perm that do not leverage the visual reconstruction, the mental image aids the jigsaw solution, proving the effectiveness of the generative approach. We observe a large performance gain for GANzzle-SA and GANzzle-VIT that take advantage of the early fusion scheme and jointly considering all patches for direct estimation of the global solution. The generative module improvement is relevant especially for PuzzleWikiArts where GANzzle struggles with the high variability of the data. The patch-oriented processing bias of GANzzle-VIT leads to a large gain in performance on puzzles of higher complexity. GANzzle-SA and GANzzle-VIT outperform other deep learning strategies. As can be noted, GANzzle-VIT bridges the gap between deep learning and optimization strategies.

We observe a similar trend for neighbor accuracy in Table 2. In general, deep learning approaches are not competitive with optimization-based strategies. The high neighbor accuracy for optimization methods reflects that the approaches tend to shift

Table 1. Comparison of direct accuracy metric on PuzzleCelebA and PuzzleWikiArts. We directly compare against deep methods [28, 34] and without mental image (Hung-perm) for comparable computational performance and include optimization methods [14, 23, 26] for complete comparison. In contrast to GANzzle strategies, Zhang et al. [34] and Hung-perm [28] are trained one model per size.

Dataset	PuzzleCelebA				PuzzleWikiArts			
	6x6	8x8	10x10	12x12	6x6	8x8	10x10	12x12
Paikin and Tal [23]	99.12	98.67	98.39	96.51	98.03	97.35	95.31	90.52
Pomeranz et al. [26]	84.59	79.43	74.80	66.43	79.23	72.64	67.70	62.13
Gallagher [14]	90.80	97.04	95.49	93.13	88.77	82.28	77.17	73.40
Zhang et al. [34]	71.96	50.12	38.05	-	12.19	5.77	3.28	-
Hung-perm [28]	33.11	12.89	4.14	2.18	8.42	3.22	1.90	1.25
GANzzle [28]	72.18	53.26	32.84	12.94	13.48	6.93	4.10	2.58
GANzzle-SA (Ours)	91.07	81.36	64.99	40.44	88.85	67.37	36.43	16.28
GANzzle-VIT (Ours)	**97.47**	**98.62**	**97.21**	**94.47**	**98.90**	**97.09**	**93.99**	**87.07**

Table 2. Results for neighbor accuracy metric on PuzzleCelebA and PuzzleWikiArts. We directly compare against deep methods [28, 34] and without mental image (Hung-perm) for similar computational performance and include optimization methods [23, 26] for complete comparison. In contrast to GANzzle strategies, Zhang et al. [34] and Hung-perm [28] are trained one model per size.

Dataset	PuzzleCelebA				PuzzleWikiArts			
	6x6	8x8	10x10	12x12	6x6	8x8	10x10	12x12
Paikin and Tal [23]	99.70	99.38	99.15	96.51	99.37	99.09	98.23	95.97
Pomeranz et al. [26]	96.31	93.87	91.38	87.79	93.39	89.96	87.25	84.07
Zhang et al. [34]	66.43	44.02	32.72	-	7.94	4.01	2.58	-
Hung-perm [28]	22.35	7.49	2.33	0.95	4.25	1.97	1.43	0.90
GANzzle [28]	66.04	46.20	26.46	9.93	11.08	7.10	5.32	4.18
GANzzle-SA (Ours)	88.43	76.96	59.08	35.52	85.69	59.28	27.96	12.68
GANzzle-VIT (Ours)	**99.35**	**98.29**	**96.61**	**93.42**	**98.75**	**96.61**	**92.90**	**84.65**

the entire puzzle of a few pieces, while keeping the global coherence of the image. The observation is in line with the edge matching heuristic that optimization approaches maximize. On the contrary, we observe lower accuracy for deep learning algorithms that are characterized by scattered erroneous assignments. GANzzle-SA and GANzzle-VIT improve over the two steps processing of patches in GANzzle and yield results on par with state-of-the-art approaches on both PuzzleCelebA and PuzzleWikiArts. Notably, GANzzle-VIT shows the effectiveness of the improvement on estimating the global solution as large drop in accuracy is not observed for larger jigsaws.

maximum of size?

The qualitative analysis in fig. 2 visualizes the estimated global solutions with different generative modules. While GANzzle recovers the spatial structure of the images on PuzzleCelebA, it struggles with the high variability of PuzzleWikiArts. On the contrary, the improved generation in GANzzle-SA and GANzzle-VIT shows a more faithful reconstruction of the target image. As can be noted, GANzzle-VIT reconstructions present higher level of details that can be exploited for later matching, e.g., less blurred hairs and shadows. We visualize jigsaw solutions in fig. 3. Despite recovering the global spatial structure, GANzzle struggles to correctly place most of the pataches. In contrast, GANzzle-SA and GANzzle-VIT achieve better placement with failure cases represented by ambiguous patches, e.g. for GANzzle-SA swapped flowers, getting close to the high accuracy of [23]. We observe improved performance on GANzzle-VIT that accurately predicts most of the input samples.

Challenging patches. We evaluate the proposed approaches on PuzzleCelebA with missing, noisy, and eroded (missing border) pieces for 6×6 puzzles in Table 3. GANzzle-VIT and GANzzle-SA benefits from the strong direct accuracy achieved when no noise is applied and improve with respect to other deep learning strategies. However, a larger drop in accuracy is observed with respect to GANzzle. GANzzle-VIT approach is robust to additive noise and struggles when erosion is applied to pieces, showing that the model learns to leverage edge information to recover the original image. Similarly to GANzzle, the proposed approaches struggle with pieces containing similar or repetitive patterns, e.g., ambiguous background patches. This limitation becomes more pronounced in puzzles with a higher degree of visual similarity among the pieces, due to the higher definition needed for the estimated solution to discriminate the pieces.

how to definite and recognize similar features is a question

Computational complexity. We compare the computational requirements of the different approaches in terms of wall time execution and memory footprint on a consumer desktop machine in Table 4. Time results are averaged over 24 samples where each jigsaw is independently solved, i.e. samples are not batched for deep learning strategies. As can be noted, optimization-based strategies [26] and [14] suffer from the time demanding execution time, especially with respect to Deep learning methods that solve the puzzle in a forward step. In contrast, Paikin and Tal [23] has comparable time requirements. Deep learning methods are largely similar, with the minimal (without GAN) Hung-Perm taking half the computational time. The proposed GANzzle-SA and GANzzle-VIT show a longer execution time with respect to the vanilla GANzzle but prove competitive with Paikin and Tal [23] and other optimization-based strategies. Further, we report the memory footprint of similarly considered solving 24 puzzles based on original authors code and respective backend environments. Results show that Deep Learning strategies reduce the time requirements at the cost of larger memory. Presented methods slightly increase the complexity of the vanilla GANzzle due to the more re-

the relationship between time and memory?
less time == increase memory?

Table 3. Comparison of missing pieces (except [23]), Gaussian noise and eroded pieces on a 6×6 puzzle for PuzzleCelebA.We directly compare against deep methods [28, 34] and without mental image (Hung-perm) for similar computational performance and include optimization methods [14, 23, 26] for complete comparison. In contrast to GANzzle strategies, Zhang et al. [34] and Hung-perm [28] are trained one model per size.

Model	Missing (%)			Noisy (σ)			Eroded (px)		
	10%	20%	30%	0.05	0.1	0.2	1	2	5
Paikin and Tal [23]	-	-	-	51.51	7.73	3.31	2.82	2.77	2.79
Pomeranz et al. [26]	52.43	24.26	25.99	87.84	89.63	91.50	6.01	16.30	15.15
Gallagher [14]	79.68	66.02	51.17	96.39	98.34	97.75	32.55	18.59	6.27
Zhang et al. [34]	64.35	60.10	**58.60**	69.87	65.30	49.85	23.81	10.93	4.84
Hung-perm [28]	29.79	26.45	23.88	31.84	29.01	21.45	25.45	26.01	9.50
GANzzle [28]	58.50	44.70	35.01	64.51	37.72	6.81	28.59	**35.47**	4.70
GANzzle-SA(Ours)	66.36	48.10	38.03	79.04	50.40	7.84	**33.33**	28.23	2.30
GANzzle-VIT(Ours)	**84.99**	**64.71**	50.58	**98.09**	**95.97**	**88.55**	32.79	26.00	**10.93**

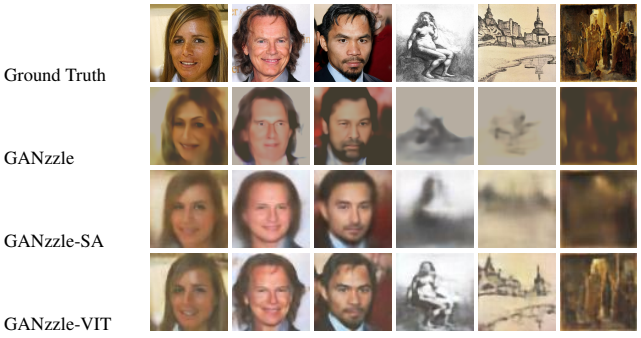

Fig. 2. Qualitative evaluation of the estimated global solution for different variant of the GANzzle strategy.

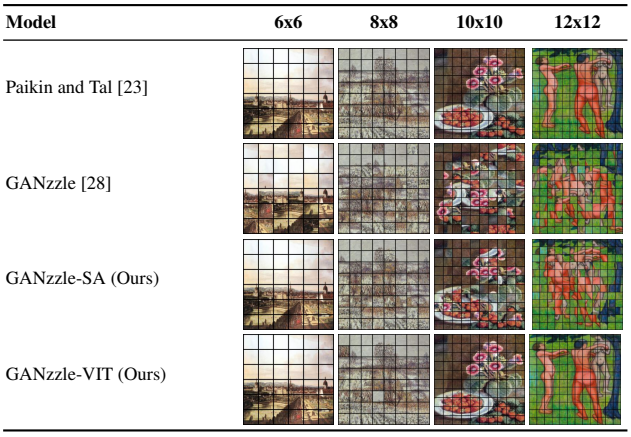

Fig. 3. Qualitative results for puzzle solving on PuzzleWikiArts with increasing complexity.

cent deep learning framework and the use of operations that are quadratic in the number of pieces.

5. Conclusions

We introduce GANzzle-SA and GANzzle-VIT, two size-agnostic puzzle solvers based on global-to-local matching of pieces. The proposed approaches address the limitations of the generative module of GANzzle achieving a large accuracy gain on open benchmark datasets such as PuzzleCelebA and PuzzleWikiArts, demonstrating the benefits of incorporating newer generative methods in the formulation. The improvement allows deep learning strategies to bridge the gap with

Table 4. Computational complexity in terms of memory footprint (RAM and VRAM in MegaBytes) and time requirements (in ms) for the different approaches on a 6×6 puzzle.

Model	RAM (MB)	VRAM (MB)	Time (ms)
Paikin and Tal [23]	715.4	-	27.47 ± 7.70
Pomeranz et al. [26]	522.1	-	221.64 ± 300.79
Gallagher [14]	525.1	-	235.19 ± 358.72
Zhang et al. [34]	2863.1	410.9	22.38 ± 8.08
Hung-Perm	2919.0	293.6	9.97 ± 1.38
GANzzle [28]	3935.1	2120.6	25.16 ± 1.1
GANzzle-SA (Ours)	3990.7	1969.0	32.58 ± 2.33
GANzzle-VIT (Ours)	4693.7	2055.6	32.66 ± 0.75

classic optimization-based approaches, which are now competitive while performing faster inference of the permutation. The fast inference has the potential to allow human-in-the-loop approaches to interactive puzzle solving, possibly over a large scale, as in the case of Frescoes, where puzzles can be seen as isolated problems and benefit from expert knowledge. Future work could explore such broken object problems in addition to shredded documents and image editing.

Limitations: As with GANzzle, the major limitation comes to the generative power of the generator. However, we have shown that improving this aspect can have significant effects on the results, allowing the approach to take advantage of state-of-the-art methods. While we acknowledge that the proposed approaches have larger computational complexity with respect to the vanilla GANzzle method due to a quadratic memory footprint, we pose that current active research on faster, efficient, and memory-friendly attention strategies will reduce the required computational complexity.

why attention model could accelerate the complex compution?how?

6. Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 964854.

References

[1] D. Bridger, D. Danon, and A. Tal. Solving jigsaw puzzles with eroded boundaries. In *CVPR*, 2020.

[2] G. Camporese, E. Izzo, and L. Ballan. Where are my neighbors? exploiting patches relations in self-supervised vision transformer. In *BMVC*, 2022.

[3] Y. Chen, X. Shen, Y. Liu, Q. Tao, and J. A. Suykens. Jigsaw-vit: Learning jigsaw puzzles in vision transformer. *Pattern Recognition Letters*, 166: 53–60, 2023.

[4] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST-8*, 2014.

[5] T. S. Cho, S. Avidan, and W. T. Freeman. The patch transform. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1489–1501, 2009.

[6] T. S. Cho, S. Avidan, and W. T. Freeman. A probabilistic image jigsaw puzzle solver. In *CVPR*, 2010.

[7] R. S. Cruz, B. Fernando, A. Cherian, and S. Gould. Visual permutation learning. In *CVPR*, 2017.

[8] A. Deever and A. Gallagher. Semi-automatic assembly of real cross-cut shredded documents. In *ICIP*, 2012.

[9] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[10] P. Dondi, L. Lombardi, and A. Setti. Dafne: A dataset of fresco fragments for digital anastlylosis. *Pattern Recognition Letters*, 2020.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[12] H. ElNaghy and L. Dorst. Complementarity-preserving fracture morphology for archaeological fragments. In *ISMM*. Springer, 2019.

[13] M. Fiorucci, M. Khoroshiltseva, M. Pontil, A. Traviglia, A. Del Bue, and S. James. Machine learning for cultural heritage: A survey. *Pattern Recognition Letters*, 2020.

[14] A. C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *CVPR*, 2012.

[15] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *ICLR*, 2019.

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

[17] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng. Jigsawgan: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks. *TIP*, 2021.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[19] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. *NeurIPS*, 2020.

[20] W. Marande and G. Burger. Mitochondrial dna as a genomic jigsaw puzzle. *Science*, 318(5849), 2007.

[21] G. Mena, D. Belanger, S. Linderman, and J. Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR*, 2018.

[22] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016.

[23] G. Paikin and A. Tal. Solving multiple square jigsaw puzzles with missing pieces. *CVPR*, 2015.

[24] G. Palmas, N. Pietroni, P. Cignoni, and R. Scopigno. A computer-assisted constraint-based system for assembling fragmented objects. In *Digital Heritage International Congress*, volume 1, pages 529–536, 2013.

[25] M.-M. Paumard, D. Picard, and H. Tabia. Deepzzle: Solving visual jigsaw puzzles with deep learning and shortest path optimization. *IEEE TIP*, 29: 3569–3581, 2020.

[26] D. Pomeranz, M. Shemesh, and O. Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *CVPR*, 2011.

[27] A. Rafique, T. Iftikhar, and N. Khan. Adversarial placement vector learning. In *ICACS*, 2019.

[28] D. Talon, A. D. Bue, and S. James. Ganzzle: Reframing jigsaw puzzle solving as a retrieval task using a generative mental image. In *ICIP*, 2022.

[29] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka. Improved artgan for conditional synthesis of natural image and artwork. *TIP*, 28(1):394–409, 2019.

[30] M. L. Teodoro, G. N. Phillips, and L. E. Kavraki. Molecular docking: A problem with thousands of degrees of freedom. In *ICRA*, volume 1. IEEE, 2001.

[31] A. van den Hengel, C. Russell, A. Dick, J. Bastian, D. Pooley, L. Fleming, and L. Agapito. Part-based modelling of compound scenes from images. In *CVPR*, 2015.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[33] T. Yu, R. Wang, J. Yan, and B. Li. Learning deep graph matching with channel-independent embedding and hungarian attention. In *ICLR*, 2019.

[34] Y. Zhang, J. Hare, and A. Prügel-Bennett. Learning representations of sets through optimized permutations. In *ICLR*, 2019.

[35] Y.-X. Zhao, M.-C. Su, Z.-L. Chou, and J. Lee. A puzzle solver and its application in speech descrambling. In *WSEAS International Conference on Computer Engineering and Applications*, pages 171–176, 2007.

In this article,GANzzle-SA and GANzzle-VIT are mentioned
to solve the puzzles in a random order in terms of global prediction,
rather than local pair.

These two models extend the GANzzle model
through adding slot attention and
visual transformer.

This is because the slot attention is multiset-equivariant
as a method of various centered-object discovery and track,
which means the order of the fragments could be ignored.
Meanwhile, as a global solution,
the visual transformer connect the different pieces
through attention mechanism.
However, this approach consume larger memory
due to large-scale dataset training and high-resolution input images.

Furthermore, the Hungarian Attention is employed to allocate the weights
and improve the accuracy of element mapping.

Their experiment is based on the dataset,"PuzzleCelebA" and "PuzzleWikiArts",
including extra test such as the missing, noisy and eroded fragments.

Compared to existing models in realm of deep learning
and optimization-based approaches,
although these two models increase memory,
they raise the accuary of direct and neighbour and save the time.