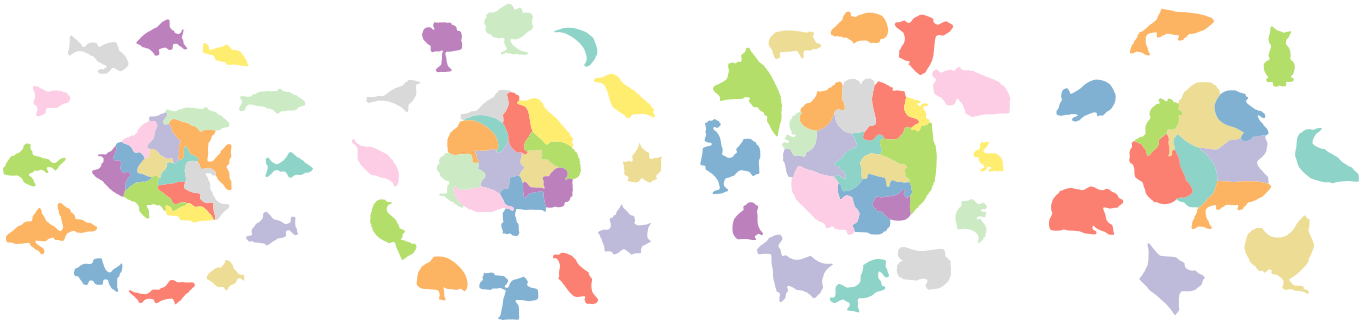
Generative Whimsy Jigsaws

JEREMY CHEW, ETH Zürich, Switzerland
KENJI TOJO, ETH Zürich, Switzerland
NOBUYUKI UMETANI, University of Tokyo, Japan
BERND BICKEL, ETH Zürich, Switzerland



Fish A (12 pieces) **Nature A** (12 pieces) **Zodiac A** (12 pieces) **Animal Tree** (7 pieces)

Fig. 1. **Generative Whimsy Jigsaws.** We present a framework to generate whimsy jigsaw puzzles—jigsaw puzzles whose pieces and overall shape depict a variety of objects through their outlines. By providing a text prompt for each piece, our framework is able to generate compelling, interconnected pieces that are guided by the textual context. The full set of prompts used for each jigsaw puzzle can be found in the supplementary material.

Unlike conventional jigsaw puzzles which partition an image into a predefined set of rectangular pieces, "whimsy" jigsaw puzzles depict a variety of objects, such as animals, vehicles, and characters, through the outlines of the pieces themselves. Although skilled artists have created visually compelling and playful puzzles within this framework, designing whimsy jigsaws remains highly challenging—creators must simultaneously partition a given outline into multiple interlocking pieces while ensuring that each piece conveys a recognizable visual concept. This difficulty limits accessibility, preventing casual creation by less experienced users, particularly when designing puzzles with many pieces and concepts. To address this challenge, we present a computational framework for automatically designing whimsy jigsaws from a loosely specified set of visual targets, each assigned to an individual piece. Specifically, we take a set of text prompts as input and use differentiable optimization to refine the geometry of each piece, guided by a pre-trained diffusion model. Since naive formulations suffer from a fundamental tension in whimsy jigsaws between geometric validity and recognizability, this paper develops a representation for planar partitioning that guarantees the partitioning of the domain into a specified number of connected pieces, while allowing for highly flexible geometric evolution. The key component is the scalar slowness field, which is used to compute a warped geodesic distance field given a point source by solving the Eikonal equation. Given distance fields from multiple sources, we can compute warped Voronoi cells that define pieces of a puzzle. The representation is fully differentiable, with gradients computed via adjoint sensitivity analysis, and the slowness field is represented as a neural implicit field. Experimental results demonstrate that our framework generates diverse, thematically consistent jigsaw puzzles across a wide range of user prompts, enabling the automated design of puzzles with complex, figurative pieces.

CCS Concepts: • **Applied computing** → **Computer-aided design**; • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: Generative design, Jigsaw puzzle generation

ACM Reference Format:
Jeremy Chew, Kenji Tojo, Nobuyuki Umetani, and Bernd Bickel. 2026. Generative Whimsy Jigsaws. In *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26), December 01–04, 2026, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3829340.3842213

1 Introduction

Creating jigsaw puzzles is generally straightforward: manufacturers typically partition an input image into puzzle pieces with predetermined shapes. These partitioning patterns ignore the image content and are often based on a regular grid, whose edges are randomly perturbed with convex and concave variations to produce interlocking pieces. Some companies, such as Wentworth Puzzles [Wentworth Puzzles 1991], Liberty Puzzles [Liberty Puzzles 2005], and Nervous System [Nervous System 2007], produce puzzles with more diverse and non-standard piece shapes. For example, some pieces are generated semi-automatically using procedural methods such as crystal growth simulation, while others are so-called "whimsy" pieces: figurative shapes that carry semantic meaning, reflecting elements of the image or the overall theme (Figure 2).

Creating whimsy jigsaw puzzles requires substantial expertise and iterative refinement. Designers must craft shapes that remain

Authors' Contact Information: Jeremy Chew, ETH Zürich, Zürich, Switzerland, jechew@ethz.ch; Kenji Tojo, ETH Zürich, Zürich, Switzerland, tojo@arch.ethz.ch; Nobuyuki Umetani, University of Tokyo, Tokyo, Japan, n.umetani@gmail.com; Bernd Bickel, ETH Zürich, Zürich, Switzerland, bickelb@ethz.ch.

This work is licensed under a Creative Commons Attribution 4.0 International License.
SA Conference Papers '26, Kuala Lumpur, Malaysia
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2842-6/2026/12
https://doi.org/10.1145/3829340.3842213



Fig. 2. **Examples of Whimsy Puzzles.** *Left to right* Cozy Cat Crowd (Image courtesy of CRAFTSMM — craftsmm.com). Photographs of whimsy jigsaw puzzles (Images courtesy of Nobuyuki Umetani, 2026).

recognizable while maintaining sufficient abstraction to ensure that pieces interlock perfectly. This makes computational design particularly challenging, as it involves balancing two conflicting requirements: adjacent pieces must share exact boundaries without gaps or overlaps, while each piece must form a recognizable and meaningful shape. While recognizability can be evaluated using recent vision-language models (VLMs), the core difficulty lies in the underlying shape representation. Explicit representations rely on heuristics and strong regularizations to avoid invalid geometry such as self-intersections and piece-wise overlaps. Hybrid approaches used in, for example, multi-material surface tracking [Heiss-Synak et al. 2024], can mitigate these issues, but introduce substantial algorithmic complexity that is hard to implement and extend. On the other hand, multi-phase implicit representations (e.g., by defining regions via multiple coupled scalar fields [Zhao et al. 1996], via their maximum [Yuan et al. 2012], or via a label function [Kim 2010]) allow flexible deformation, but do not guarantee that each piece forms a single connected component. Our proposed parameterization, which uses a generalized Voronoi formulation, guarantees connectivity and geometrical validity, while admitting flexible deformation without the need for heuristics, strong regularizations, or specialized machinery.

The Voronoi diagram is a spatial partition of the input region into multiple cells, where each point in a cell is nearest to its site. This has desirable properties: the cells naturally fill the input region without gaps or overlaps, and furthermore, each cell is simply connected, corresponding to a single site within the cell. While traditional Voronoi cells are restricted to convex polygons under the Euclidean distance, we enhance their expressiveness with a slowness field. The slowness field defines a spatially-varying distance metric—a path experiences a longer distance in regions of high slowness. By manipulating this slowness field, we can design jigsaw puzzles as Voronoi cells, where each piece has sufficient expressiveness to convey the desired semantics.

We represent the slowness field as a neural implicit function to ensure a compact yet flexible representation. Then, we numerically solve the Eikonal equation on a regular grid to compute the Voronoi diagram. This process is fully differentiable with respect to the parameters of the neural implicit function. To guide the puzzle pieces based on a provided text prompt, we rely on a pretrained diffusion model [StabilityAI 2023]. Specifically, we utilize score distillation sampling (SDS) [Poole et al. 2022] to optimize the shapes to match a pre-specified textual concept. The optimization is also augmented with several regularization losses to improve the quality of the pieces. Optionally, we can texture the puzzle pieces as a post-processing step using a VLM. Our formulation bridges the gap between geometric validity and semantic expressiveness.

We evaluate our method across a variety of themes, including human faces, animals, and aquatic creatures. In many examples, both individual puzzle pieces and the overall assembly shape carry semantic meaning. We also validate our approach through ablation studies, including comparisons with explicit shape representations. The puzzles are physically fabricable, as we show in several examples. The code is available on Github[1].

2 Related Work

2.1 Puzzle Design

Several works have addressed the design of jigsaw puzzles via computational methods. For example, Lau et al. [2014] present a method for optimizing puzzle cuts that conform to the contours of a provided image and user-defined curves, generating interlocking pieces with custom outlines. Elber and Kim [2022] create jigsaw pieces over freeform 2-manifolds, lifting jigsaw puzzles from planar domains to curved 3D surfaces. These methods design puzzles based on pre-defined pieces or user-provided outlines. In contrast, our work explores 2D *shapes* of jigsaw puzzle pieces, and how the individual pieces can be optimized to convey a specific theme.

Other works explore various types of related puzzle design. Kita and Miyata [2021] design 2D and 3D polyomino-based jigsaw puzzles, but require an already-voxelized input shape, and also lack semantic control over the shape of the pieces. Relaxing the voxelization requirement, Duncan et al. [2017] find freeform pieces that can be rearranged to form close approximations of two target shapes. In a similar vein, Kita [2023] finds polyomino-dissections between two shapes with color information, though at the cost of once again requiring a grid-based discretization.

In 3D, freeform pieces are much harder to generate. Instead, Song et al. [2012], Tang et al. [2019], and Chen et al. [2022] design three-dimensional interlocking puzzles, relying significantly on a voxelized model.

2.2 Tilings and Packings

Jigsaw puzzles are closely related to packing problems [Dowsland and Dowsland 1992], where methods are found to pack objects as densely as possible without overlaps. These problems have a wide variety of applications (see Sweeney and Paternoster [1992], Lodi et al. [2002], and Wäscher et al. [2007] for surveys), but we mostly focus on two main bodies of work: image collages and mosaics—problems that try to pack multiple different pieces into a specified

[1] https://github.com/mickey1356/generative_jigsaws

container, and tilings—problems that pack multiple copies of some preset shapes *without gaps*.

Image Collages and Mosaics. These works look at the packing of various images and objects in a specified container in a visually-appealing manner. Jigsaw Image Mosaics [Kim and Pellacini 2002] use an energy-based framework for mosaic problems, generating a global tessellation constraint and then deforming the primitives to fit them within the cells. Despite its name, the mosaics still exhibit small gaps in between pieces. Avoiding expensive global constraints, Kwan et al. [2016] propose a shape descriptor that is locally supported, enabling efficient partial-shape matching. Alternatively, Wang and Lu [2025] use a differentiable renderer to explore the problem in image space, also avoiding the need for global constraints.

Focusing instead on visual appeal, [Huang et al. 2011] look at generating Arcimboldo-like collages with images obtained from the Internet, allowing for thematically-related pieces. Our works also uses thematically-related pieces, but relies on pre-trained VLMs [Radford et al. 2021; Rombach et al. 2022] to generate new objects instead. Xu and Kaplan [2007], and later, Zou et al. [2016], explore calligraphic packing, focusing on the specific case where the objects to be packed are letters, thus requiring certain legibility constraints. Finally, Yang et al. [2022] approach the problem from the perspective of human visual perception, using Gestalt psychology principles to generate visually-pleasing collages.

Tiling Problems. Unlike image collages and mosaics, tiling problems require that objects are packed with no gaps. One example is Escherization [Kaplan and Salesin 2000], where the authors morph a given image such that it perfectly tiles the infinite 2D plane. Similar works also try to tackle this problem, presenting better fitting methods [Koizumi and Sugihara 2011], or introducing a deformable mesh [Nagata and Imahori 2021]. Some other works focus on designing tiles for tiling a surface [Chen et al. 2023; Eigensatz et al. 2010; Fu et al. 2010; Singh and Schaefer 2010], and some target the inverse problem, finding a set of tiles sufficient to cover a flexibly-discretized 2D domain [Chen et al. 2025]. Recently, machine learning models [Xu et al. 2020] have also been used to solve the tiling problem without the need of ground-truth tiling solutions.

Perhaps the work most closely aligned to ours is Generative Escher Meshes [Aigerman and Groueix 2024]. The authors optimize the geometry of symmetric tiles guided by a text prompt. Additionally, they include the ability to encode multiple prompts within a single tile, not dissimilar to a jigsaw puzzle. However, they note that doing so leads to less plausible results. Furthermore, the connectivity of the jigsaw pieces has to be pre-defined. While our generative jigsaws have no constraints with tileability, we can optimize for a larger number of pieces, and the connectivity of our pieces is also more flexible.

2.3 Voronoi Diagrams

Voronoi diagrams have been used for a wide variety of applications, including fluid and foam simulation [de Goes et al. 2015; Numerow et al. 2024], structural design and optimization [Feng et al. 2023; Martínez et al. 2019], cosmological simulations [Springel 2010], additive manufacturing [Martínez et al. 2016], and geometric design [Burch et al. 2020; Polat and İlerisoy 2020; Wu et al. 2024]. Although several works have generalized Voronoi diagrams to non-Euclidean distance metrics [Martínez et al. 2019; Nair 2025; Zhou et al. 2025], existing methods are restricted to simple analytic metrics or star-shaped cells, limiting their flexibility in accommodating diverse geometric designs. In contrast, we introduce a slowness-field representation that enables flexible control over geodesics, yielding a substantially wider range of spatially-varying distance metrics and non-convex cell shapes, and enabling flexible design of complex, semantically-driven puzzle shapes. Geodesic Voronoi diagrams have been extensively studied in the context of *forward* geodesic computation [Kimmel and Sethian 1998; Mitchell et al. 1987; Peyré and Cohen 2003; Zhao 2004]. Instead, we tackle the orthogonal problem of *inversely* optimizing distance metrics to achieve semantically meaningful cell boundary shapes.

2.4 Vision-Language Models

The introduction of CLIP [Radford et al. 2021], a latent-space coupling between text and images, enabled guided optimization of pixels for image stylization [Kwon and Ye 2022], vector strokes for sketch generation [Vinker et al. 2023, 2022], and even 3D models [Gao et al. 2023; Khalid et al. 2022; Michel et al. 2021; Sanghi et al. 2022]. Later, the development of diffusion models [Nichol et al. 2022; Ramesh et al. 2022; Rombach et al. 2022] allowed for higher-fidelity and more realistic text-guided image generation.

Score distillation [Poole et al. 2022; Wang et al. 2022] was then introduced as a method to optimize 3D NeRFs [Mildenhall et al. 2020] for text-to-3D generation. It has since been used for a wide range of different modalities, including visual illusions [Burgert et al. 2023; Geng et al. 2024], vector art [Jain et al. 2022], typography [Iluz et al. 2023], low-poly stylization [Wiersma et al. 2026], wire art [Tojo et al. 2024], and even Escher tiles [Aigerman and Groueix 2024]. In our work, we also utilize the strong image and text priors provided by a pre-trained diffusion model (DeepFloyd-IF [StabilityAI 2023]) for optimizing the geometry of jigsaw puzzles and their pieces.

3 Method

We assume that we are given a set of text prompts that describe both the individual pieces and the overall shape of the puzzle. Some prompts may be omitted, allowing the pieces to take on arbitrary shapes instead. Our goal, then, is to produce a jigsaw puzzle whose pieces and overall shape visually match the provided texts. In this regard, we note three key constraints:

(i) *Piece-wise* **Each** piece is a single connected region
(ii) *Puzzle-wise* **All** pieces should together form one single connected region
(iii) *Jigsaw-fit* Adjacent pieces should have a jigsaw-like fit (i.e., they should exactly share common edges with no gaps in between)

Our approach, as mentioned, is based on the Voronoi diagram, which satisfies both topological consistency and geometric validity. The Voronoi diagram of a 2D point set $\mathcal{P} = \{\mathbf{p}_s\}_{s=1}^{S}, \mathbf{p}_s \in \mathbb{R}^2$ can be obtained by expanding multiple circular wave fronts from each point $\mathbf{p}_s$ (i.e., site) and drawing a boundary where they collide. Mathematically, the diagram can be computed by solving S Eikonal

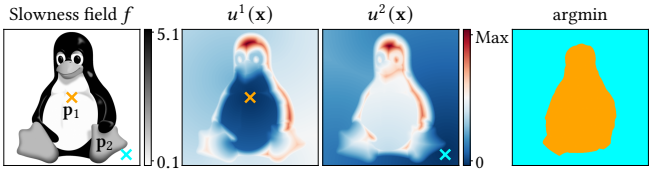


Fig. 3. **Spatially-varying Eikonal Equation.** We allow the slowness field f to vary across the domain (here, we use an image of Tux [Ewing 1996], where black represents areas of high slowness) and observe how the resultant time field is strongly dependent on f as well as the source locations. After taking the argmin, this splits the domain into interconnected regions which are all defined by f.

equations:

$$|\nabla u^s(\mathbf{x})| = f(\mathbf{x}), \tag{1}$$

with boundary conditions $u^s(\mathbf{p}_s) = 0$, and $f : \mathbb{R}^2 \to (0, \infty)$ is a positive scalar field that describes the inverse of the velocity (i.e., slowness) of the wavefront at a given position $\mathbf{x}$. We call f the *slowness field*. The solution $u^s : \mathbb{R}^2 \to \mathbb{R}$ can then be interpreted as the minimal time required for the wavefront to arrive at $\mathbf{x}$ from $\mathbf{p}_s$, which we call the *arrival time field*.

Note that when $f(\mathbf{x}) = 1$, the solution u^s is the Euclidean signed distance field from $\mathbf{p}_s$. Then, the point $\mathbf{x}$ belongs to the Voronoi cell $s(\mathbf{x})$ that minimizes the arrival time across all arrival-time fields u^s, i.e. $s(\mathbf{x}) = \arg\min_s u^s(\mathbf{x})$. However, the resulting Voronoi cells from a constant slowness field are convex and lack the flexibility to align with the textual prompts.

One option is to take inspiration from Wu et al. [2024] and assign multiple Voronoi cells for a single piece. However, this either requires a large number of cells for more complex shapes or requires accepting pieces made up of long, straight lines. Another option is to use an alternative metric for the Voronoi diagram, in the vein of Martínez et al. [2019] and Zhou et al. [2025], but their star-shaped metrics limit the types of resulting geometries.

Therefore, we present a generalized Voronoi formulation that supports flexible non-convex partitions. Our key idea is to control the speed of the wavefront propagations by optimizing the slowness field f, allowing the slowness to vary spatially across the domain (Figure 3). The slowness field f is a single, global parameter that directly affects the shape of the Voronoi partitions, while satisfying constraints (i) and (iii) by construction. We enforce constraint (ii) manually by adding a separate surrounding border, and we prove the per-cell connectedness and the exact-boundary properties in the supplementary material.

3.1 Differentiable Pipeline

In order to solve the Eikonal equation, we first discretize the 2D domain into $N \times N$ grid cells. Then, we use the fast sweeping method (FSM) [Zhao 2004] implemented on a GPU to obtain the numerical solutions u^s_{ij} for each source point $\mathbf{p}_s$ where $i, j \in \{1, \ldots, N\}$. While the typical fast sweeping method sweeps over the grid in four directions, our implementation instead updates all grid cells in parallel without enforcing directional traversal orders. Iterating this relaxation procedure for $4N$ iterations is sufficient to achieve convergence [Zhao 2004].

For the cell shape optimization, we seek to compute the gradient of the arrival time field at a grid point u^s_{ij} with respect to the input parameters, namely the slowness field f and the source-point locations $\mathbf{p}_s$. To this end, we employ the adjoint method to back-propagate the gradients of u^s through the Eikonal solver. Similar to the forward pass, we update all grid cells in parallel without directional sweeps, iterating for $4N$ iterations to achieve convergence. During each update, we identify which neighboring cells contributed to the current value and propagate gradients accordingly. Pseudo-code for both passes can be found in the supplementary material.

We then employ a softmin relaxation with a temperature β to normalize u^s_{ij} as

$$\tilde{u}^s_{i,j} = \frac{\exp(-\beta u^s_{i,j})}{\sum_{s'} \exp(-\beta u^{s'}_{i,j})}. \tag{2}$$

The values $\tilde{u}^s_{i,j} \in [0, 1]$ define a probability distribution over the puzzle pieces, indicating the likelihood that pixel (i, j) belongs to piece s. Unlike the discrete argmin operator, the smoothness of softmin enables gradient flow back to all the arrival time fields $u^1, \ldots, u^S$, improving optimization stability. By progressively increasing β, we can anneal the soft assignment toward a discrete partition, recovering the discrete argmin operator in the limit $\beta \to \infty$.

The resultant arrival time field u^s from a given source position $\mathbf{p}_s$ corresponds to piece s of the whole jigsaw puzzle. Since the normalized arrival time field has values that lie between 0 and 1, we can treat each field as a grayscale image. To combine pieces, we simply sum the normalized arrival time fields involved, which can also be interpreted as a grayscale image.

In order to guide the generated images toward the text prompts, we employ the SDS loss first proposed by Poole et al. [2022]. At each iteration, the optimized image x is perturbed with Gaussian noise to form $x_t = \alpha_t x + \sigma_t \varepsilon$. A pre-trained UNet model [Saharia et al. 2022] then attempts to predict the noise, and the final loss is defined by the gradients of the original diffusion loss:

$$\nabla_\theta \mathcal{L}_{SDS} = w(t)\,\{\varepsilon_\theta(x_t, t, y) - \varepsilon\}\,\frac{\partial x}{\partial \theta}, \tag{3}$$

where y is the text embedding of the given prompt produced by a pre-trained CLIP text encoder [Radford et al. 2021], θ the image parameters, and $w(t)$ a constant multiplier that depends on α_t.

3.2 Optimization

We now have all the components necessary for our optimization pipeline (Figure 4). We assume that we are given S, the number of requested pieces, as well as the textual context $c_{i,s}$ for the individual pieces and c_o for the overall jigsaw shape. We note that these textual contexts are optional, and, if omitted, will simply result in the SDS loss ignoring those specific pieces.

We begin by initializing $S + 8$ source points. The first S source points $\mathbf{p}_s$ correspond to the jigsaw pieces, while the remaining 8 source points are marked as **background pieces**. This is necessary because the partitions obtained by solving the Eikonal equations fill the entire $N \times N$ grid, thereby limiting the combined shape to a

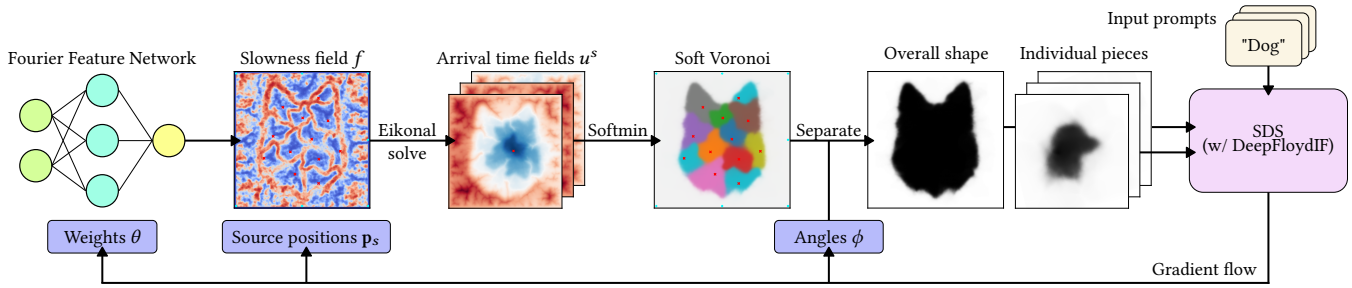


Fig. 4. **Overview.** Given the input prompts (in yellow), we optimize the weights of a Fourier feature network (FFN), the source positions, and the angles (all in purple). At each iteration, the FFN is used to generate the slowness field, which, after an Eikonal solve, gives us arrival time fields for each source position. We then apply the softmin to obtain a soft Voronoi diagram, and extract the overall shape and the individual pieces. The pieces are preprocessed (cropped and rotated), and all the images are sent to a **frozen** pre-trained diffusion model (in pink) that drives the shapes to convey the given prompts.

square. By introducing background pieces, we significantly expand the geometries the overall jigsaw shape can take.

We consider the domain of the grid to be $[-1, 1]^2$. The $\mathbf{p}_s$ are initialized using a Fibonacci lattice on a disk centered at $(0, 0)$ with a radius of 0.7, are set to be **optimizable**, and are clamped to lie within the domain during optimization. While random initializations based on seed are also possible, and, in our experiments, do not lead to any significant qualitative difference, we opt for a Fibonacci lattice. This ensures the points are roughly evenly spread [Saff and Kuijlaars 1997], giving each piece an initial partition of roughly equal area.

The 8 background points are set to be the four corners and the center of the four sides of the unit square, but inset by 0.03 units. Unlike $\mathbf{p}_s$, these background points are **fixed** and **non-optimizable**.

We draw inspiration from Tancik et al. [2020] and parameterize f using Fourier feature networks (FFNs). This allows f to converge much faster while learning high-frequency features that are otherwise difficult with a direct parameterization using the image pixels. Furthermore, this representation is resolution-independent, allowing us to obtain high-resolution versions of f after the optimization is completed. Given the input point $\mathbf{v}$ in pixel coordinates, and a random Gaussian matrix $\mathbf{B} \in \mathbb{R}^{F\times 2}$, the mapping is given by

$$\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]^T \ , \tag{4}$$

where F is the number of Fourier features.

Specifically, we use $F = 128$ and pass $\gamma(\mathbf{v})$ into a multi-layer perceptron (MLP) with 3 hidden layers, each with 256 nodes. The results are then passed through the sigmoid function, which we interpret as f. For optimization, we use a resolution of $N = 512$ pixels.

We then linearly map f into the appropriate range of values for the slowness field, where $f_{\min} = 0.1$ and $f_{\max} = 5.1$. Additionally, after obtaining f, to ensure the background pieces form a "barrier" preventing the jigsaw pieces from touching the sides, we manually set an outer pixel border of 8 pixels to be $f_{\min}$. As mentioned previously, this also enforces constraint (ii), forcing the jigsaw puzzle to remain as a single connected region.

The next few steps of the pipeline use the components described in the previous section. Namely, we solve S Eikonal equations, and compute their softmin. Before passing the results into the SDS loss, however, we perform a small pre-processing step that is detached from the gradient computation. In this step, we compute a rough bounding box of the silhouette (obtained by thresholding the softmin with a value of 0.5), pad it to form a square, and scale it back up using nearest-neighbor interpolation to its original dimensions. Then, we rotate the scaled image by a random, **optimizable**, angle ϕ_s and pass the final augmented result into the SDS loss. This allows for the generation of non-upright silhouettes, which the diffusion models appear to avoid.

We use DeepFloyd-IF-L [StabilityAI 2023] as the underlying diffusion model for the SDS loss. DeepFloyd-IF works in pixel space, which we find gives higher-quality results and faster runtimes than other diffusion models which work in latent space. We use a classifier-free guidance value of 10, and a timestep range of 2% to 98%.

We also introduce several regularization losses. We apply a smoothness loss on f following Alexa and Matusik [2010] that minimizes the mean-squared Laplacian of each pixel in f:

$$\mathcal{L}_{smooth} = \frac{1}{N^2}\sum_{i,j}\left(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j}\right)^2 \ . \tag{5}$$

We then add a size loss that aims to balance the area of each piece a_s, estimated by computing the mean of the pixel values over u^s. This loss is computed as

$$\mathcal{L}_{size} = \sum_{s=1}^{S}\left[a_s \log a_s S - \frac{k_1}{N^2}\sum_{i,j} u^s_{i,j}\log u^s_{i,j}\right] , \tag{6}$$

where $k_1 = 0.05$ is a weighting constant that was found through qualitative experimentation.

Finally, we apply a regularization loss on f itself. Intuitively, a small f results in larger areas, as each source is able to expand more quickly. This loss is made up of two terms. The first term, an area-weighted penalty, encourages f to be small in the immediate vicinity of a source point. It adds a minimum-radius circle of low f (represented by a 2D Gaussian distribution) around each source point, forcing each piece to fill its immediate region, encouraging a minimum area to prevent "dying" pieces that shrink to the seeds.

The second term applies a uniform pressure to keep f small:

$$\mathcal{L}_{regular} = \sum_{s=1}^{S} \frac{1}{N^2} \sum_{i,j} G_{\mathbf{p}_s,r}(i,j)\, f_{i,j}\, a_s^{-1} + k_2 \sum_{i,j} f_{i,j}\,. \tag{7}$$

$G_{\mathbf{p}_s,r}(x,y)$ is a 2D Gaussian centered at $\mathbf{p}_s$ with radius r, and k_2 is a preset weighting constant. We use $r = 0.2$ and $k_2 = 0.1$.

Our final loss term is then given by

$$\mathcal{L} = w_o \mathcal{L}_{SDS}(I_o, c_o) + \frac{w_i}{S} \sum_{s=1}^{S} \mathcal{L}_{SDS}(I_{i,s}, c_{i,s}) + w_{smooth}\mathcal{L}_{smooth} + w_{size}\mathcal{L}_{size} + w_{regular}\mathcal{L}_{regular}\,, \tag{8}$$

where I_o is the overall image, $I_{i,s}$ are the individual jigsaw pieces, and the various w's are preset weights of the various loss terms. We set $w_o = 10$, $w_{smooth} = 20$, $w_{size} = 500$, and $w_{regular} = 50$. For w_i, we schedule it following the sigmoid function from 20 to 30 with a warm-up period of 500 iterations.

We run each optimization for 5000 iterations. We use the AdamW [Loshchilov and Hutter 2019] optimizer to optimize f, p_s, and the rotation angles ϕ_s. The three parameters have initial learning rates of 5×10^{-4}, 10^{-4}, and 10^{-4} respectively, and each learning rate is scheduled in a manner similar to Poole et al. [2022], with a linear warm-up for 2000 iterations and a cosine decay for the remaining iterations. We also schedule the temperature of the softmin following the sigmoid function from 10 to 100, with a warm-up period of 2000 iterations. We note that even though $\mathbf{p}_s$ and ϕ_s are set to be optimizable, in practice, their values remain mostly constant throughout the optimization. This is likely due to their low weight and small influence in the final shapes. Indeed, perturbing the source positions in areas of low slowness bounded by high slowness hardly changes the final shape.

3.3 Post-processing

Once the optimization is finished, we convert the jigsaw puzzle into its explicit polygonal representation. We first extract a high-resolution version of the slowness map (1024×1024 pixels). We then run the FSM to obtain the arrival time fields, and apply the discrete argmin to obtain the Voronoi-like partitions. For each grid cell, we create a vertex at its four corners, and add edges between these vertices if the neighboring cells belong to different pieces. Finally, we perform iterative Laplacian smoothing [Sorkine et al. 2004] for 1000 iterations with a kernel size of 2 and a learning rate of 2×10^{-3} to remove grid artifacts, and a line simplification algorithm [Douglas and Peucker 1973; Ramer 1972] with a threshold of 10^{-4} to reduce the number of vertices and edges.

4 Results

We now present our main results, ablations, and comparisons. Our differentiable Eikonal solver is implemented in CUDA/C++ and exposed to Python using nanobind [Jakob 2022]. The rest of the pipeline is implemented in PyTorch. Each jigsaw puzzle takes between 25–50 minutes for 5000 iterations on a single NVIDIA RTX 4090 GPU, depending on the number of pieces (we generate puzzles with 6 to 20 pieces).

4.1 Validation

Convergence. We first validate our relaxation of the fast sweeping method, where we update all grid cells in parallel without enforcing directional traversal orders. We iterate this relaxation procedure for $4N$ iterations, which, as we show in Figure 5, is more than sufficient to achieve convergence. For this experiment, we define a slowness field f defined as $f(x,y) = 5\sin^2(2\pi xy) + 0.1$. We set the source point $\mathbf{s}$ to be $(0,0)$. For grid sizes $N \in \{128, 256, 512, 1024\}$, we first compute u_{ref}, using a typical implementation of FSM with directional traversal orders. Then, we use our relaxation to compute u_k at various iteration counts k up to $8N$. We plot the error, given by the infinity norm $\|u_k - u_{\text{ref}}\|_\infty$, against k. At all four resolutions, convergence to the reference solution is achieved at around $1.5N$, leaving us a roughly threefold margin at $4N$.

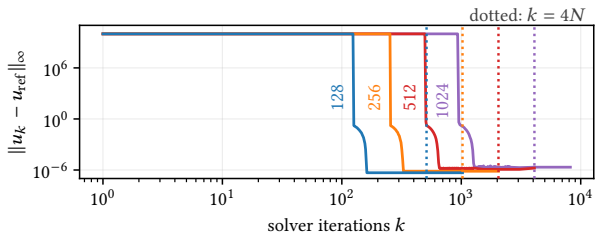


Fig. 5. **Convergence of Random-order FSM.** For grid sizes $N \in \{128, 256, 512, 1024\}$ (curves labelled by N), we plot $\|u_k - u_{\text{ref}}\|_\infty$ against the iteration count k. For roughly the first N iterations, the error is dominated by cells yet unreached by the source, which still hold their initial value. It then drops sharply, reaching convergence (within floating-point limits) well within the preset iteration limit of $4N$ (dotted).

Gradients. We then validate the gradients of our differentiable Eikonal solver. We use the same slowness field f defined previously. This time, however, we set the source point $\mathbf{s}$ to be $(0.1, 0.2)$ instead, and add a simple loss function $\mathcal{L}(u) = \sum_{m,n} u$. We compute gradients using the adjoint method described in Section 3.1 and compare them with gradients obtained using finite differences with a step size of 10^{-4} (Figure 6). We also compute the gradients with respect to the source positions (i.e. $\partial_{\mathbf{s}_{x,y}} \mathcal{L}$) and compare against finite differences with the same step size. The two gradients are nearly identical, demonstrating that we can correctly compute gradients of the Eikonal solver with respect to the slowness field.

4.2 Experiments

Whimsy Jigsaws. Figures 1, 12, and 13 show the results that we generated using our method. We generate jigsaw puzzles over a wide variety of prompts, from animals to nature to man-made objects. We also generate several thematic puzzles, including the Chinese zodiac and country-specific ones. Despite the competing geometric constraints of each piece, our method still produces compelling silhouettes and puzzles with interesting shapes.

Textures. Although the silhouette alone can carry enough semantic meaning to be recognizable, using textures (such as in Figure 2) can convey additional information, turning even the most abstract

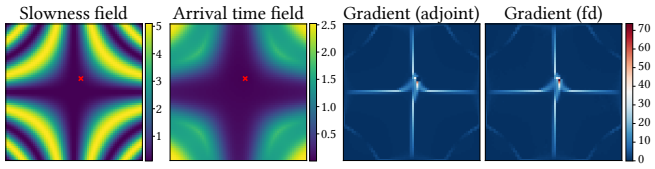


Fig. 6. **Gradients of Eikonal Solver.** We compare the gradients computed using the adjoint method (Left) to those computed using finite differences (Right), for the specified slowness field and loss function. Note that the magnitude scales for both gradient plots are equal. For the position gradients, we obtained gradients of [341.2243, 252.2181] using the adjoint method, and [341.5318, 251.8585] using finite differences (relative error: 0.1%).

silhouette into an easily understandable object. We first triangulate the jigsaw puzzle pieces using Triangle [Shewchuk 1996] to compute a UV mapping for each piece. We pass the UV maps into a Vision-Language Model (VLM) (we use Google Gemini [Google 2025] here) to generate the corresponding textures (Figure 7). Please refer to the supplementary material for the full prompts.

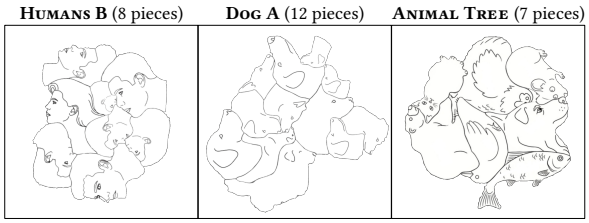


Fig. 7. **Textured Puzzles.** To further enhance the semantic meaning of the pieces, we generate textures using a VLM and apply them to the jigsaw puzzle.

Physical Prototypes. As jigsaw puzzles are inherently tactile toys, we physically manufacture several of our generated jigsaw puzzles as shown in Figure 10. We 3D print the 7-piece **ANIMAL TREE**, as well as the 15-piece **CAT B**. Additionally, we use a laser cutter to cut and engrave the textured 12-piece **DOG A** and the 12-piece **FISH A**.

Failure Cases. We present a few examples of failure cases that we encounter in Figure 8. Due to the stochastic nature of SDS gradients, the overall assembled shape may fail to reflect the target prompt and instead collapse into nearly circular shapes. Furthermore, because we rely heavily on the underlying diffusion model, we face similar limitations [Rombach et al. 2022], including an inability to generate thin, elongated features. Additionally, despite our regularization losses, the way the source locations are clustered and shifted during optimization may cause some pieces to "die out" (i.e., shrink significantly). However, these failure cases are not common. In a run of 20 differently-seeded experiments with the same prompt ("fish"), we observed only 2 cases where the overall shape collapsed to a circle, and 1 case where "dying out" occurred. In practice, a random restart usually mitigates these issues.

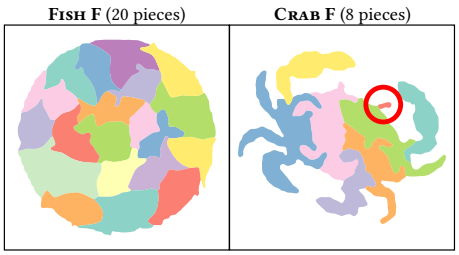


Fig. 8. **Failure Cases.** *Left*: The optimization (given the prompt "fish" for **both** the individual pieces and the assembled shape) collapses the overall shape into a circle instead of a fish. The individual pieces remain unaffected by this collapse. *Right*: The optimization (given the prompt "crab" for both the pieces and the overall shape) is unable to generate the elongated legs of the crab (visible in the orange and green pieces), and also exhibits a piece "dying out" (circled).

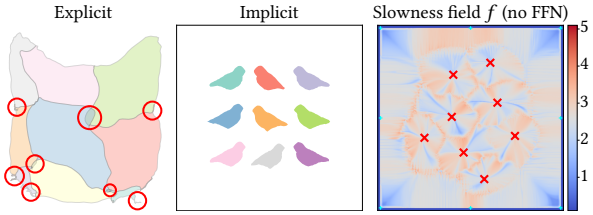


Fig. 9. **Ablations.** *Left*: Explicit representations result in invalid geometry, such as self-intersections, and overlapping pieces. *Middle*: Implicit representations allow for flexible deformation, but lack connectivity guarantees. *Right*: Removing the FFN parameterization loses the high-frequency features needed for compelling silhouettes.

4.3 Ablations

We evaluate the effect of our design choices in Figure 9. Specifically, we ablate our two key parameterization choices: the jigsaw pieces and the slowness field.

In the first experiment, we use an explicit representation of the jigsaw puzzle pieces as a set of polylines, initializing a 3×3 grid of squares, where each edge is made up of 32 segments of equal length. Then, we render the lines using DiffVG [Li et al. 2020] before passing the rendered image to the SDS loss. We also include a smoothness loss that minimizes the squared Laplacian of of the polylines for each piece. Although we use an extremely high weight (1000) to penalize self-intersections, this representation remains prone to invalid geometry and self-intersections and lacks the flexibility provided by our proposed parameterization.

In the second experiment, we use an implicit representation inspired by multi-phase flow([Zhao et al. 1996]), where each piece is represented as the zero level-set of an optimizable scalar field. We partition the union into disjoint regions by assigning each point to the shape it lies deepest inside, rasterize them using a sigmoid function, and pass them to the SDS loss. While this shares similarities with our parameterization, allowing for flexible deformation, it lacks the connectivity guarantees necessary for a jigsaw puzzle.

Lastly, we avoid using an FFN to parameterize the slowness field and directly optimize the pixel values of the field. Consistent with

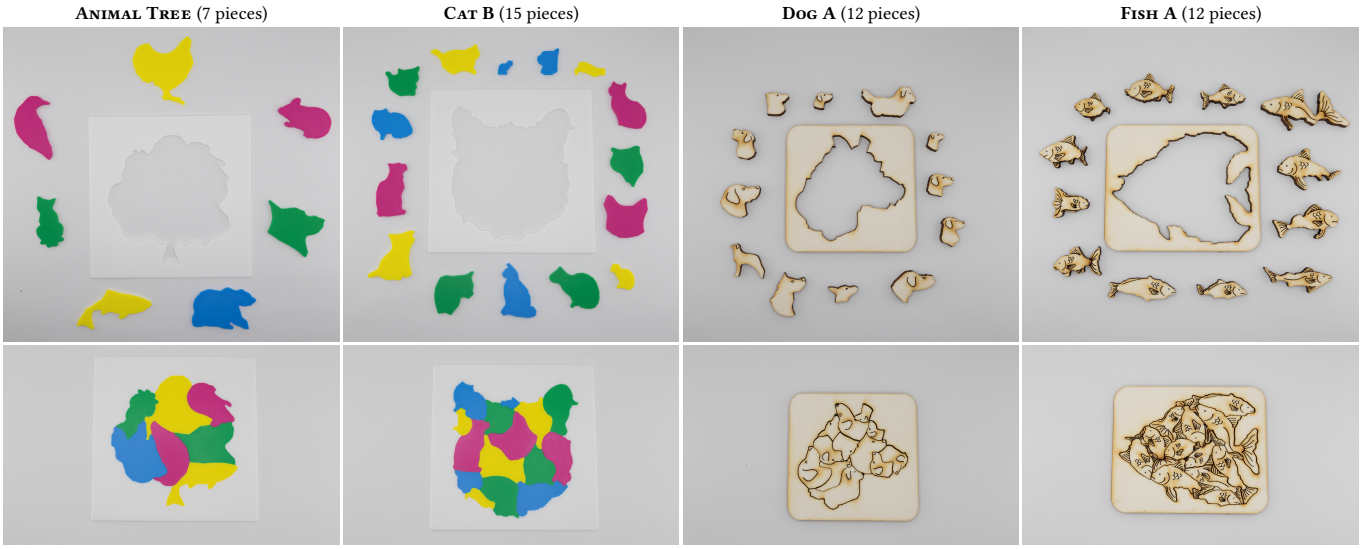


Fig. 10. **Physical Prototypes.** We 3D print and laser cut selected samples of our jigsaw puzzles.

the observations of Tancik et al. [2020], we find that, without the FFN parameterization, the slowness field struggles to capture high-frequency variations, directly resulting in less plausible results.

4.4 Comparisons

As there are no existing methods that directly address the same task, we compare against close alternatives. The first uses the "Split tiles" option presented in Generative Escher Meshes [Aigerman and Groueix 2024]. However, after adapting their code to our setting, we were unable to obtain results comparable in quality to those reported in their paper. We therefore provide a qualitative comparison with their reported results and refer readers to their work for further details. Their framework is also limited in the number of pieces it can optimize (they mention that plausibility degrades fairly quickly).

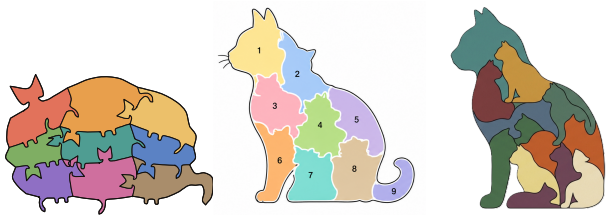


Fig. 11. **VLM Generation.** We prompt VLMs in an attempt to generate a whimsy jigsaw puzzle. (*Left*: Claude, *Middle*: ChatGPT, *Right*: Claude)

The second alternative (Figure 11) involves prompting various VLMs (Google Gemini and Claude) using a description of the jigsaw puzzle we desire (please refer to the supplementary material for the full prompt used). However, this approach offers limited geometric control, and while some pieces appear whimsical, they lack the same consistency provided by our method. Overall the quality of the pieces provided by the VLMs is lower than our method, and the overall shapes generated are usually more abstract and less recognizable.

5 Conclusion

We present a method to design whimsy jigsaw puzzles. Beyond being a novelty toy, we believe this also has applications in art, architecture, and interior design, enabling the creation of unique and innovative designs for tiling and artworks.

Additionally, our parameterization can be further extended by lifting it into 3D, as none of the key steps are restricted to the 2D plane. We can then represent each "piece" as a three-dimensional signed distance field, and compute the corresponding Voronoi diagram over the 3D domain instead. Coupled with existing text-to-3D methods [Poole et al. 2022], one could then possibly design sculptures made up of smaller, semantically-related, sculptures.

Our method is not without its limitations, however. Firstly, the number of pieces that our framework is able to generate is still fairly small (9 to 15) compared to commercial puzzles, which has upwards of 1000 pieces. While our method is still technically-sound even with larger number of pieces, the competing geometry and limited resolution (of the SDS loss) lead to degenerate pieces and less visually-appealing results. Nevertheless, we believe that our puzzles can carve out a niche in the area of artisanal puzzles—smaller puzzles that consist of highly-intricate pieces.

Our method uses the SDS loss, which relies on the underlying diffusion model to guide the shapes. As such, we inherit its existing limitations. For example, our framework works best with specific, concrete, concepts, and may fail to converge properly when given generic and abstract ones. Additionally, we lack exact granular control over the shape of the pieces and the jigsaw puzzle. However,

since the diffusion model can be treated as a black-box for associating images with a text-based loss, we can efficiently swap it out for future methods which perform better.

Last but not least, as seen in Figure 7, texturing the jigsaw puzzle pieces could enhance recognizability. Currently, we rely on a VLM to generate textures as an optional post-processing step, entirely decoupled from shape optimization. One could instead perform joint texture-shape optimization similar to Generative Escher Meshes [Aigerman and Groueix 2024]. However, this requires careful tuning: the texture loss can be satisfied by painting a background texture to produce a spurious outline of a piece rather than having the foreground object's texture span the full geometry, effectively decoupling the piece's boundary shape from the object it was meant to depict. This lets the overall shape drift toward arbitrary geometry (as seen in several of the pieces of the Cozy Cats Crowd puzzle in Figure 2), rather than preserving a semantically relevant outline. As such, we leave this avenue, ensuring that both shape and texture convey similar semantic meaning, for future work.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. The second author was also supported in part by JSPS KAKENHI Grant Number 23KJ0699 and by JST ASPIRE Grant Number JPMJAP2401.

References

Noam Aigerman and Thibault Groueix. 2024. Generative Escher Meshes. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3641519.3657452

Marc Alexa and Wojciech Matusik. 2010. Reliefs as Images. *ACM Transactions on Graphics* 29, 4 (July 2010), 60:1–60:7. doi:10.1145/1778765.1778797

Michael Burch, John van Lith, Nick van de Waterlaat, and Jurrien van Winden. 2020. Voronoier: from images to Voronoi diagrams. In *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction (VINCI '20)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3430036.3430043

Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael S. Ryoo. 2023. Diffusion Illusions: Hiding Images in Plain Sight. doi:10.48550/arXiv.2312.03817 arXiv:2312.03817 [cs].

Rulin Chen, Xuyang Ma, Praveer Tewari, Chi-Wing Fu, and Peng Song. 2025. Inverse Tiling of 2D Finite Domains. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3757377.3763834

Rulin Chen, Pengyun Qiu, Peng Song, Bailin Deng, Ziqi Wang, and Ying He. 2023. Masonry Shell Structures with Discrete Equivalence Classes. *ACM Trans. Graph.* 42, 4 (July 2023), 115:1–115:12. doi:10.1145/3592095

Rulin Chen, Ziqi Wang, Peng Song, and Bernd Bickel. 2022. Computational design of high-level interlocking puzzles. *ACM Trans. Graph.* 41, 4 (July 2022), 150:1–150:15. doi:10.1145/3528223.3530071

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power particles: an incompressible fluid solver based on power diagrams. *ACM Transactions on Graphics (TOG)* 34, 4 (July 2015), 50:1–50:11. doi:10.1145/2766901

David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica* 10, 2 (Dec. 1973), 112–122. doi:10.3138/FM57-6770-U75U-7727

Kathryn A. Dowsland and William B. Dowsland. 1992. Packing problems. *European Journal of Operational Research* 56, 1 (Jan. 1992), 2–14. doi:10.1016/0377-2217(92)90288-K

Noah Duncan, Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. 2017. Approximate dissections. *ACM Trans. Graph.* 36, 6 (Nov. 2017), 182:1–182:13. doi:10.1145/3130800.3130831

Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graph.* 29, 4 (July 2010), 45:1–45:10. doi:10.1145/1778765.1778782

Gershon Elber and Myung-Soo Kim. 2022. Synthesis of 3D jigsaw puzzles over freeform 2-manifolds. *Computers & Graphics* 102 (Feb. 2022), 339–348. doi:10.1016/j.cag.2021.10.014

Larry Ewing. 1996. Tux.

Fan Feng, Shiying Xiong, Ziyue Liu, Zangyueyang Xian, Yuqing Zhou, Hiroki Kobayashi, Atsushi Kawamoto, Tsuyoshi Nomura, and Bo Zhu. 2023. Cellular topology optimization on differentiable Voronoi diagrams. *Internat. J. Numer. Methods Engrg.* 124, 1 (2023), 282–304. doi:10.1002/nme.7121

Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. 2010. K-set tilable surfaces. *ACM Trans. Graph.* 29, 4 (July 2010), 44:1–44:6. doi:10.1145/1778765.1778781

William Gao, Noam Aigerman, Thibault Groueix, Vladimir G. Kim, and Rana Hanocka. 2023. TextDeformer: Geometry Manipulation using Text Guidance. doi:10.48550/arXiv.2304.13348 arXiv:2304.13348 [cs].

Daniel Geng, Inbum Park, and Andrew Owens. 2024. Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 24154–24163. doi:10.1109/CVPR52733.2024.02280

Google. 2025. Gemini: A Family of Highly Capable Multimodal Models. doi:10.48550/arXiv.2312.11805 arXiv:2312.11805 [cs]

Peter Heiss-Synak, Aleksei Kalinov, Malina Strugaru, Arian Etemadi, Huidong Yang, and Chris Wojtan. 2024. Multi-Material Mesh-Based Surface Tracking with Implicit Topology Changes. *ACM Trans. Graph.* 43, 4 (July 2024), 54:1–54:14. doi:10.1145/3658223

Hua Huang, Lei Zhang, and Hong-Chao Zhang. 2011. Arcimboldo-like collage using internet images. In *Proceedings of the 2011 SIGGRAPH Asia Conference (SA '11)*. Association for Computing Machinery, New York, NY, USA, 1–8. doi:10.1145/2024156.2024189

Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. 2023. Word-As-Image for Semantic Typography. *ACM Trans. Graph.* 42, 4 (July 2023), 151:1–151:11. doi:10.1145/3592123

Ajay Jain, Amber Xie, and Pieter Abbeel. 2022. VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models. doi:10.48550/arXiv.2211.11319 arXiv:2211.11319 [cs]

Wenzel Jakob. 2022. nanobind: tiny and efficient C++/Python bindings. https://github.com/wjakob/nanobind.

Craig S. Kaplan and David H. Salesin. 2000. Escherization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 499–510. doi:10.1145/344779.345022

Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. 2022. CLIP-Mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*. 1–8. doi:10.1145/3550469.3555392 arXiv:2203.13333 [cs].

Byungmoon Kim. 2010. Multi-phase fluid simulations using regional level sets. *ACM Trans. Graph.* 29, 6, Article 175 (Dec. 2010), 8 pages. doi:10.1145/1882261.1866197

Junhwan Kim and Fabio Pellacini. 2002. Jigsaw Image Mosaics. *ACM Trans. Graph.* 21, 3 (July 2002), 657–664. doi:10.1145/566654.566633

R. Kimmel and J. A. Sethian. 1998. Computing Geodesic Paths on Manifolds. *Proceedings of the National Academy of Sciences of the United States of America* 95, 15 (1998), 8431–8435. https://www.jstor.org/stable/45775

Naoki Kita. 2023. Dissection Puzzles Composed of Multicolor Polyominoes. *Computer Graphics Forum* 42, 7 (2023), e14968. doi:10.1111/cgf.14968 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14968.

Naoki Kita and Kazunori Miyata. 2021. Computational design of polyomino puzzles. *The Visual Computer* 37, 4 (April 2021), 777–787. doi:10.1007/s00371-020-01968-5

Hiroshi Koizumi and Kokichi Sugihara. 2011. Maximum Eigenvalue Problem for Escherization. *Graphs and Combinatorics* 27, 3 (May 2011), 431–439. doi:10.1007/s00373-011-1022-5

Kin Chung Kwan, Lok Tsun Sinn, Chu Han, Tien-Tsin Wong, and Chi-Wing Fu. 2016. Pyramid of arclength descriptor for generating collage of shapes. *ACM Trans. Graph.* 35, 6 (Dec. 2016), 229:1–229:12. doi:10.1145/2980179.2980234

Gihyun Kwon and Jong Chul Ye. 2022. CLIPstyler: Image Style Transfer with a Single Text Condition. doi:10.48550/arXiv.2112.00374 arXiv:2112.00374 [cs].

Cheryl Lau, Yuliy Schwartzburg, Appu Shaji, Zahra Sadeghipoor, and Sabine Süsstrunk. 2014. Creating Personalized Jigsaw Puzzles. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering (NPAR '14)*. Association for Computing Machinery, New York, NY, USA, 31–39. doi:10.1145/2630397.2630405

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.* 39, 6 (Nov. 2020), 193:1–193:15. doi:10.1145/3414685.3417871

Liberty Puzzles. 2005. Liberty Puzzles. https://libertypuzzles.com/. Retrieved on 2026-04-21.

Andrea Lodi, Silvano Martello, and Michele Monaci. 2002. Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141, 2 (Sept. 2002), 241–252. doi:10.1016/S0377-2217(02)00123-6

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. doi:10.48550/arXiv.1711.05101 arXiv:1711.05101 [cs]

Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural voronoi foams for additive manufacturing. *ACM Transactions on Graphics (TOG)* 35, 4 (July 2016),

44:1–44:12. doi:10.1145/2897824.2925922

Jonàs Martínez, Mélina Skouras, Christian Schumacher, Samuel Hornus, Sylvain Lefebvre, and Bernhard Thomaszewski. 2019. Star-shaped metrics for mechanical metamaterial design. *ACM Trans. Graph.* 38, 4 (July 2019), 82:1–82:13. doi:10.1145/3306346.3322989

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2021. Text2Mesh: Text-Driven Neural Stylization for Meshes. doi:10.48550/arXiv.2112.03221 arXiv:2112.03221 [cs].

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. doi:10.48550/arXiv.2003.08934 arXiv:2003.08934 [cs].

Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. 1987. The Discrete Geodesic Problem. *SIAM J. Comput.* 16, 4 (Aug. 1987), 647–668. doi:10.1137/0216045

Yuichi Nagata and Shinji Imahori. 2021. Escherization with Large Deformations Based on As-Rigid-As-Possible Shape Modeling. *ACM Trans. Graph.* 41, 2 (Nov. 2021), 11:1–11:16. doi:10.1145/3487017

Vishnu G. Nair. 2025. Metric-Driven Voronoi Diagrams: A Comprehensive Mathematical Framework. *Computation* 13, 9 (Sept. 2025), 212. doi:10.3390/computation13090212

Nervous System. 2007. Nervous System. https://n-e-r-v-o-u-s.com/index.php. Retrieved on 2026-04-21.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. doi:10.48550/arXiv.2112.10741 arXiv:2112.10741 [cs].

Logan Numerow, Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2024. Differentiable Voronoi Diagrams for Simulation of Cell-Based Mechanical Systems. *ACM Trans. Graph.* 43, 4 (July 2024), 69:1–69:11. doi:10.1145/3658152

Gabriel Peyré and Laurent D. Cohen. 2003. Geodesic Remeshing Using Front Propagation. In *Proceeding of VLSM'03*. Nice, France, 33–40. https://hal.science/hal-00365627

Helin Polat and Zeynep Yeşim İlerisoy. 2020. A Geometric Method on Facade Form Design with Voronoi Diagram. *Modular Journal* 3, 2 (Dec. 2020), 179–194. https://dergipark.org.tr/en/pub/modular/article/788953

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D Using 2D Diffusion. doi:10.48550/arXiv.2209.14988 arXiv:2209.14988 [cs]

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. https://arxiv.org/abs/2103.00020v1.

Urs Ramer. 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing* 1, 3 (Nov. 1972), 244–256. doi:10.1016/S0146-664X(72)80017-0

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. doi:10.48550/arXiv.2204.06125 arXiv:2204.06125 [cs].

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. doi:10.48550/arXiv.2112.10752 arXiv:2112.10752 [cs].

Edward. B. Saff and Arno. B. J. Kuijlaars. 1997. Distributing Many Points on a Sphere. *The Mathematical Intelligencer* 19, 1 (Dec. 1997), 5–11. doi:10.1007/BF03024331

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. doi:10.48550/arXiv.2205.11487 arXiv:2205.11487 [cs]

Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. 2022. CLIP-Forge: Towards Zero-Shot Text-to-Shape Generation. doi:10.48550/arXiv.2110.02624 arXiv:2110.02624 [cs].

Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Selected papers from the Workshop on Applied Computational Geormetry, Towards Geometric Engineering (FCRC '96/WACG '96)*. Springer-Verlag, Berlin, Heidelberg, 203–222.

Mayank Singh and Scott Schaefer. 2010. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph.* 29, 4 (July 2010), 46:1–46:7. doi:10.1145/1778765.1778783

Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. 2012. Recursive interlocking puzzles. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 128:1–128:10. doi:10.1145/2366145.2366147

O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (SGP '04)*. Association for Computing Machinery, New York, NY, USA, 175–184. doi:10.1145/1057432.1057456

Volker Springel. 2010. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society* 401, 2 (Jan. 2010), 791–851. doi:10.1111/j.1365-2966.2009.15715.x arXiv:0901.4107 [astro-ph.CO].

StabilityAI. 2023. DeepFloyd IF: a novel state-of-the-art open-source text-to-image model with a high degree of photorealism and language understanding. https://github.com/deep-floyd/IF. Retrieved on 2026-04-20.

Paul E. Sweeney and Elizabeth Ridenour Paternoster. 1992. Cutting and Packing Problems: A Categorized, Application-Orientated Research Bibliography. *Journal of the Operational Research Society* 43, 7 (July 1992), 691–706. doi:10.1057/jors.1992.101 _eprint: https://doi.org/10.1057/jors.1992.101.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. doi:10.48550/arXiv.2006.10739 arXiv:2006.10739 [cs]

Keke Tang, Peng Song, Xiaofei Wang, Bailin Deng, Chi-Wing Fu, and Ligang Liu. 2019. Computational Design of Steady 3D Dissection Puzzles. *Computer Graphics Forum* 38, 2 (2019), 291–303. doi:10.1111/cgf.13638 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13638.

Kenji Tojo, Ariel Shamir, Bernd Bickel, and Nobuyuki Umetani. 2024. Fabricable 3D Wire Art. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3641519.3657453

Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. 2023. CLIPascene: Scene Sketching with Different Types and Levels of Abstraction. doi:10.48550/arXiv.2211.17256 arXiv:2211.17256 [cs].

Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. CLIPasso: semantically-aware object sketching. *ACM Trans. Graph.* 41, 4 (July 2022), 86:1–86:11. doi:10.1145/3528223.3530068

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. 2022. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. doi:10.48550/arXiv.2212.00774 arXiv:2212.00774 [cs].

Zhenyu Wang and Min Lu. 2025. Image-Space Collage and Packing with Differentiable Rendering. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3721238.3730690

Wentworth Puzzles. 1991. Wentworth Wooden Jigsaw Company. https://www.wentworthpuzzles.com/. Retrieved on 2026-04-21.

Ruben Wiersma, Alexandre Binninger, Peizhuo Li, Tanguy Magne, Annika Oehri, Aviv Segall, Danielle Luterbacher, Marcel Padilla, Jing Ren, and Olga Sorkine-Hornung. 2026. Gradient Descent in the ALPS: Abstracted Low-Poly Stylization and Fabrication. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3799902.3811050

Xuanyu Wu, Kenji Tojo, and Nobuyuki Umetani. 2024. Free-Form Floor Plan Design Using Differentiable Voronoi Diagram. *Pacific Graphics Conference Papers and Posters* (2024). doi:10.2312/PG.20241296

Gerhard Wäscher, Heike Haußner, and Holger Schumann. 2007. An improved typology of cutting and packing problems. *European Journal of Operational Research* 183, 3 (Dec. 2007), 1109–1130. doi:10.1016/j.ejor.2005.12.047

Hao Xu, Ka-Hei Hui, Chi-Wing Fu, and Hao Zhang. 2020. TilinGNN: learning to tile with self-supervised graph neural network. *ACM Trans. Graph.* 39, 4 (Aug. 2020), 129:129:1–129:129:16. doi:10.1145/3386569.3392380

Jie Xu and Craig S. Kaplan. 2007. Calligraphic packing. In *Proceedings of Graphics Interface 2007 (GI '07)*. Association for Computing Machinery, New York, NY, USA, 43–50. doi:10.1145/1268517.1268527

Zuyi Yang, Qinghui Dai, and Junsong Zhang. 2022. Visual perception driven collage synthesis. *Computational Visual Media* 8, 1 (March 2022), 79–91. doi:10.1007/s41095-021-0226-8

Zhan Yuan, Yizhou Yu, and Wenping Wang. 2012. Object-space multiphase implicit functions. *ACM Trans. Graph.* 31, 4, Article 114 (July 2012), 10 pages. doi:10.1145/2185520.2185610

Hongkai Zhao. 2004. A Fast Sweeping Method for Eikonal Equations. *Math. Comp.* 74, 250 (May 2004), 603–627. doi:10.1090/S0025-5718-04-01678-3

Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. 1996. A Variational Level Set Approach to Multiphase Motion. *J. Comput. Phys.* 127, 1 (Aug. 1996), 179–195. doi:10.1006/jcph.1996.0167

Haoyang Zhou, Logan Numerow, Stelian Coros, and Bernhard Thomaszewski. 2025. Closed-Form Construction of Voronoi Diagrams with Star-Shaped Metrics. *ACM Trans. Graph.* 44, 6 (Dec. 2025), 236:1–236:13. doi:10.1145/3763296

Changqing Zou, Junjie Cao, Warunika Ranaweera, Ibraheem Alhashim, Ping Tan, Alla Sheffer, and Hao Zhang. 2016. Legible compact calligrams. *ACM Trans. Graph.* 35, 4 (July 2016), 122:1–122:12. doi:10.1145/2897824.2925887

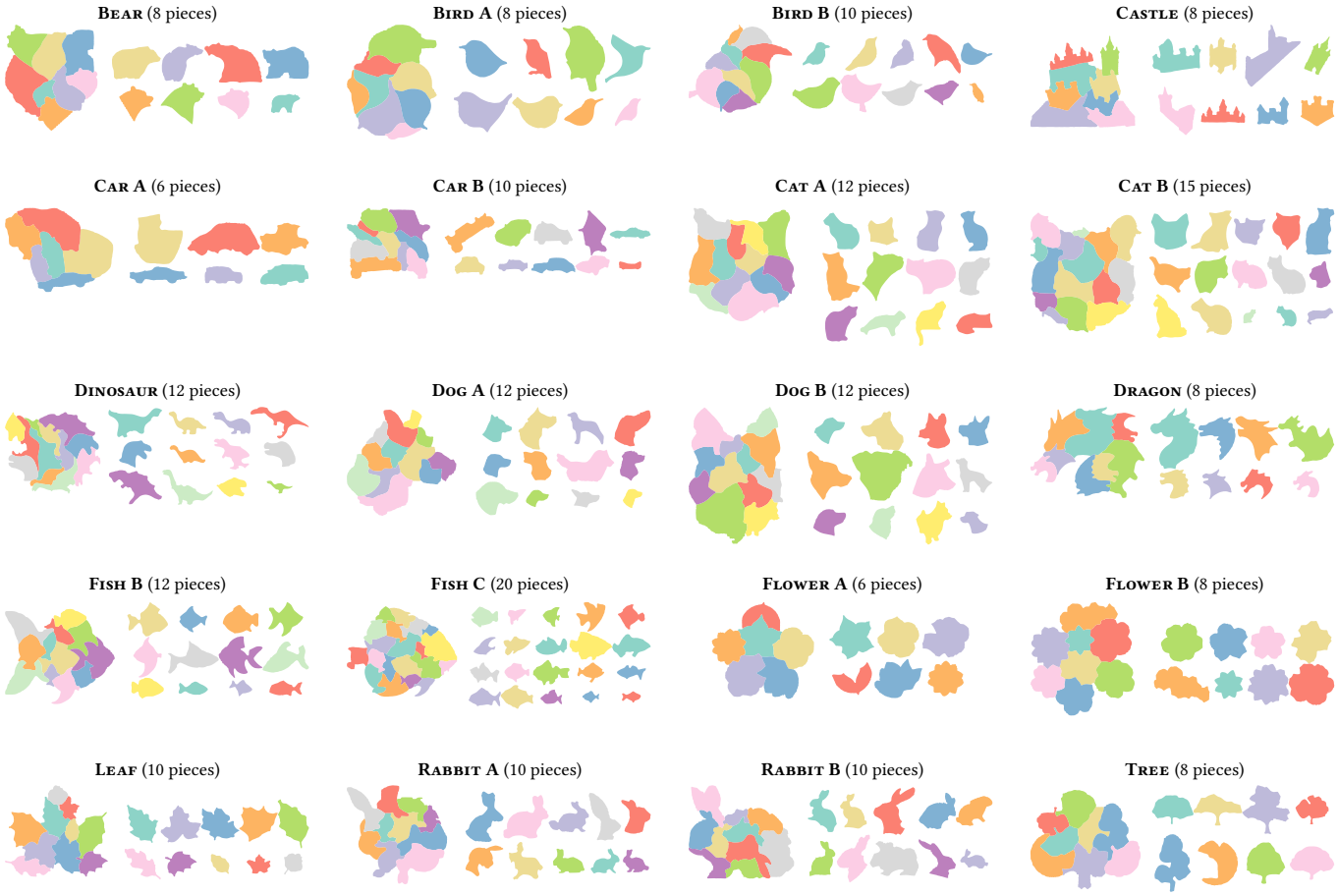


Fig. 12. **Gallery (Single Prompt).** We show examples of whimsy jigsaw puzzles generated by our method. These puzzles use the **same** prompt for each individual piece, as well as the overall shape. For the full prompts, please refer to the supplementary material.

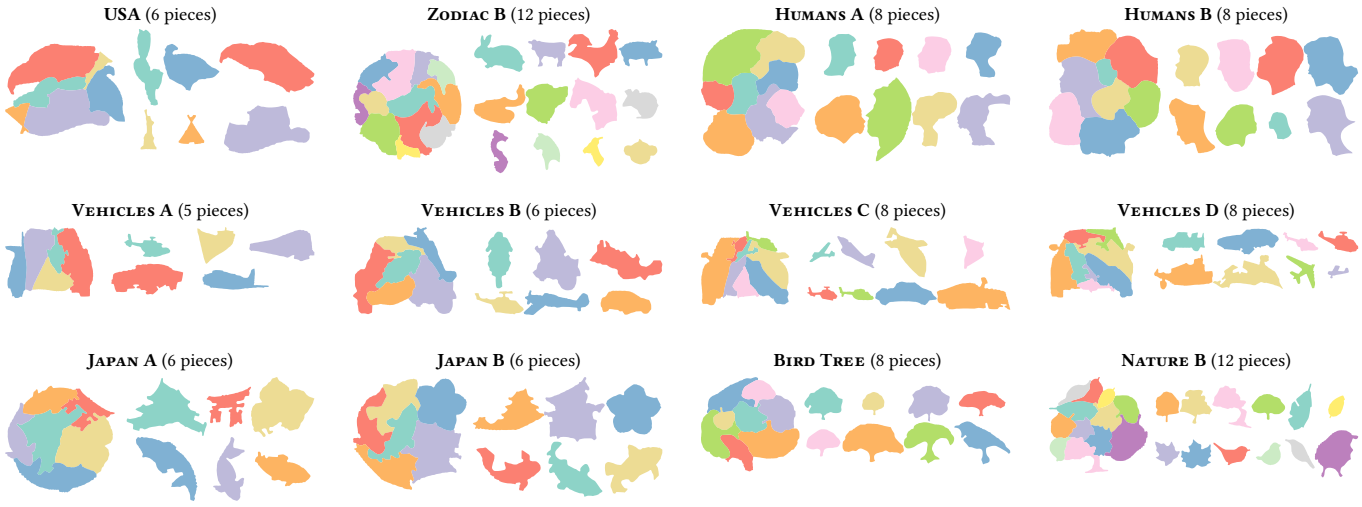


Fig. 13. **Gallery (Multi-prompt).** We show examples of whimsy jigsaw puzzles generated by our method. These puzzles use **different** prompts for the pieces and the overall shape. Again, for the full prompts, please refer to the supplementary material.