applied sciences

MDPI

Article

PatMamba: Grid-Aware Vision Mamba for Compound Patent Figure Segmentation

Zheng Ma and Feng Zhang *

Library, China University of Geosciences (Wuhan), Wuhan 430074, China; mazheng@cug.edu.cn
* Correspondence: zhangfeng@cug.edu.cn

Abstract

Patent documents express much of their technical content in drawings, yet a typical compound figure packs several sub-figures together with a dense overlay of reference numerals and lead lines, so downstream tasks such as prior art search and figure-text association must first decompose the figure and separate the annotation text from the line-art. Existing methods treat this as a binary "sub-figure versus background" segmentation that discards the text as noise and encodes no structural prior. We propose PatMamba, a Vision Mamba framework built on a VMamba SS2D backbone and a U-Net decoder that jointly predicts three semantic classes, namely background, sub-figure regions and text-label regions. PatMamba also incorporates a grid-aware scan that appends two grid row/column-priority directions to the four cross-scan directions, thereby encoding the compound-figure layout into the selective-scan sequence at negligible parameter cost. On the DeepPatent2 benchmark, the Mamba backbone lifts figure-level sub-figure detection from 70.7% for the traditional run-length-smoothing baseline to 94.5% hit@IoU$_{0.7}$, ahead of the CNN (72.6%) and Transformer (93.7%) backbones; the grid-aware scan lifts the pixel-level mIoU to 0.918 and yields its largest gains on dense compound figures with four or more sub-figures, improving the sub-figure IoU by +1.3 and +1.7 points and the Boundary F1 by +0.7 points. In zero-shot transfer to utility patents, biomedical compound figures, chemical reaction schemes and handwritten circuit diagrams, this structure-aware bias persists on the visually closest domain and decays in a regular gradient with domain distance.

Keywords: state space model; selective scan; compound figure decomposition; reference numeral separation; sub-figure detection; layout analysis

1. Introduction

1.1. Background and Motivation

Patents are among the most detailed and rigorously codified records of technological progress, and for a large class of inventions the decisive technical content is carried by the drawings rather than by the prose. A patent figure is very rarely a single image: it is typically a compound figure that assembles several sub-figures, such as perspective views, cross-sections, exploded assemblies, flow charts and detail enlargements, onto one sheet, and overlays them with a dense network of reference numerals, letters and lead lines that connect each annotation to the component it identifies [1,2]. These annotations are not decorative; they are the formal mechanism by which the written description refers to the drawings, since every "element 102" named in the claims points to the numeral "102" drawn on the figure. The annotation text is, therefore, superimposed directly on, and

Check for updates

Academic Editor: Panayiotis Vlamos

Received: 17 October 2026
Revised: 17 November 2026
Accepted: 18 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

frequently across, the boundaries between sub-figures, so that the graphical and the textual content of a patent are physically entangled in a single raster image.

Any system that consumes patent figures must first resolve this entanglement: at every pixel it must decide whether that pixel belongs to a sub-figure region, to a text-label region, or to the background. This decomposition is the gateway to an entire chain of patent intelligence tasks. Prior art search and infringement comparison require individual sub-figures to be localized and aligned with their counterparts in other documents; technology-function-effect matrices and figure-text association require each reference numeral to be matched to the sub-figure it annotates and to the sentence of the description that explains it [3,4]. Retrieval, classification and description systems built on large vision-language models are increasingly applied to patent figures, but their usefulness is bounded by the quality of the underlying decomposition: a model that cannot reliably tell a numeral from a line cannot reason reliably about the figure it is asked to describe [5]. Robust, pixel-accurate segmentation of compound patent figures is, therefore, not an auxiliary preprocessing step but a foundational layer on which patent figure understanding is built.

The difficulty of this task is specific to technical line-art. Patent drawings are essentially binary, white background, black line images whose informational content is carried by thin black strokes, and adjacent sub-figures are separated by narrow gaps that are easily lost at low resolution. Unlike natural photographs, they offer no texture, color or shading to disambiguate regions and unlike printed documents, the "text" and the "figure" share the same stroke vocabulary and are deliberately superimposed. Reference numerals are frequently placed inside a sub-figure or straddling its boundary, so a naive text stripping step removes exactly the pixels that define the sub-figure outline. Furthermore, the layout of a compound figure is itself meaningful: sub-figures are arranged on an implicit grid, in row-major order with occasional irregular panels, and a method that respects this grid is expected to be more robust than one that treats the image as an unstructured field of pixels. These properties make compound patent figures a demanding, large-scale and industrially relevant segmentation target, and they motivate both the task formulation and the architecture proposed in this paper.

These challenges are not unique to patents: scientific papers, technical reports and textbooks exhibit the same compound-figure structure, and recent work has begun to model it directly, for instance by aligning sub-figures with their captions [6]. What motivates a patent-focused treatment is a combination of scale, legal semantics and drawing normativity. Patents form one of the largest and most uniform collections of compound technical figures in existence, and their reference numerals are the formal mechanism by which the claims refer to the drawings, so that mis-separating a numeral from its sub-figure becomes a claim-level error rather than a cosmetic one. Patent drawings are also governed by formal drafting rules that mandate reference numbering, lead lines and specific line work conventions, so the text-figure entanglement is systematic and legally binding rather than incidental and the layout is itself normative, an embodiment being conventionally drawn as a row-major arrangement of views that matches the grid structure our scan exploits. The method is, therefore, not limited to patents in principle, because the grid-aware scan should transfer to any grid-organized compound figure. Nevertheless, patents are the setting where the decomposition is most consequential and where the largest labeled corpus exists, and we benchmark there accordingly.

1.2. Related Gaps

Three lines of prior work bear directly on this problem, and each leaves a gap that this paper addresses. The first is patent figure segmentation itself. Existing formulations cast the problem as a binary "sub-figure versus background" segmentation in which the annotation

text is treated as noise to be removed rather than as a first-class semantic category [7]. In that setting, the reference numerals are either discarded or recovered afterward by a disconnected heuristic, and the two objectives of clean sub-figure boundaries and clean text separation are never reconciled within a single optimization. As our experiments show (Section 4), a network trained only to separate sub-figures from background learns to blur the boundaries wherever numerals lie, because from the binary viewpoint those pixels are ambiguous; the consequence is that downstream sub-figure detection degrades even when an aggregate pixel metric still looks acceptable.

The second line is reference numeral extraction, which has historically been a standalone, rule-based preprocessing stage: optical character recognition over connected components, run-length smoothing to merge the numeral strokes into blobs, and hand-crafted geometric filters to discard non-text strokes [8]; patent-specific applications of this pipeline for drawing text detection are described in [9,10]. Such pipelines are order-dependent and error-prone: an OCR failure or a mis-merged blob propagates directly into the sub-figure boundary estimate, because the text removal step and the boundary optimization step never communicate. More recent benchmarks push toward fine-grained detection and OCR of the component identifiers themselves [11], but they still treat the text as an isolated object rather than as part of a joint segmentation objective. Our formulation instead promotes the text-label regions to an explicit third semantic class of the segmentation network, so that text separation and sub-figure segmentation are learned jointly and consistently.

The third line is document layout analysis, which separates a scanned page into text blocks, figures, tables and other regions [12]. This literature descends only to the granularity of the figure block: it determines where a figure lies on the page, but not how the figure is composed internally. The interior of a compound figure, where sub-figures meet and where numerals overlap strokes, is precisely the region of interest for patent analysis, yet it is left unaddressed by page-level methods.

Two further, orthogonal gaps concern the modeling side. To the best of our knowledge, no prior work simultaneously (i) predicts text labels as a pixel-level semantic class inside the segmentation network and (ii) injects the compound-figure grid layout as an explicit structural prior. In addition, the fine boundaries between sub-figures are easily destroyed by the aggressive downsampling of standard encoders, and the input resolutions commonly used in natural image segmentation are inadequate for resolving the narrow gaps between adjacent panels. This calls for an encoder with a global receptive field, so that a sub-figure boundary can be reasoned about in the context of the whole sheet, while still retaining linear complexity so that high-resolution inputs remain tractable. State-space models offer exactly this trade-off and have not, to our knowledge, been brought to bear on technical drawing segmentation. Recent 2025–2026 work on compound-figure analysis, which we build upon, reinforces rather than closes this gap: vision-language methods for patent and scientific figures [3–5] consume an already decomposed figure, in which the sub-figures are already separated from their labels, to perform captioning, retrieval or question answering. They assume the very decomposition whose quality this paper addresses, and none of them learns the pixel-level joint segmentation of sub-figure and text-label regions with an explicit layout prior.

1.3. Contributions

Our contributions are fivefold.

1. New task and benchmark. We formulate the first pixel-level three-class segmentation of compound patent figures, namely background, sub-figure regions and text-label regions and provide the ground-truth construction from the DeepPatent2 annotations

together with a patent-level 60/20/20 split that prevents figures of the same patent from leaking across splits.

2. Grid-aware scan. We introduce a grid-aware scan that appends two grid row/column-priority directions to the four cross-scan directions of the VMamba SS2D block. This explicitly encodes the compound-figure grid structure into the selective-scan sequence modeling, at negligible parameter and complexity cost, and concentrates its gains on dense multi-panel figures.
3. Backbone adaptation for technical line-art. We adapt the Vision Mamba backbone with a U-Net decoder as a drop-in alternative to CNN and Transformer backbones for high-resolution technical drawings, where linear-complexity global context and dense skip connections are both essential for resolving fine sub-figure boundaries and the narrow gaps between adjacent panels.
4. Systematic comparison and figure-level detection. We report the first systematic comparison of a traditional run-length-smoothing baseline, a CNN (U-Net), a Transformer (SegFormer) and Mamba models on DeepPatent2, using both pixel-level segmentation metrics and the figure-level sub-figure detection metric aligned with prior work.
5. Cross-domain benchmark corpus. We compile and release E2-300, a set of 300 US utility patent compound figures with sub-figure ground truth, which together with three external corpora (biomedical, chemical and circuit figures) supports the zero-shot cross-domain evaluation of Section 4.7.

1.4. Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews related work on patent figure segmentation, reference numeral extraction, document layout analysis, vision state-space models and scan direction strategies and positions our contribution relative to each. Section 3 presents the proposed PatMamba framework, including the SS2D backbone, the grid-aware scan module, the decoder and the loss. Section 4 describes the dataset, the evaluation protocol and the experimental results, including ablations that localize the effect of the grid-aware scan and a zero-shot cross-domain study over four external corpora. Section 5 discusses the findings, the metric alignment with prior work and the limitations. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Patent Figure Segmentation

The closest prior work is the segmentation of technical drawing figures in patents, which sits at the intersection of document analysis and instance segmentation. The DeepPatent corpus established large-scale patent drawing recognition and retrieval [1], and DeepPatent2 extended this into a benchmarking corpus for technical drawing understanding with figure-level and component-level annotations [2]. On this substrate, Hoque et al. framed the sub-figure detection problem and compared a series of segmentation backbones, notably the Medical Transformer (MedT) [13], DETR [14], U-Net [15] and HRNet [16]. Their formulation and evaluation protocol are the reference point for our own, and we deliberately report our detection numbers on the same figure-level box-IoU footing so that the two sets of results remain comparable [7].

The 99.5% figure-level accuracy reported for MedT is an instance-level, two-class (sub-figure versus background) figure hit rate measured at box-IoU thresholds of 0.7 and 0.9 over roughly 4000 figures. Our task is a pixel-level, three-class segmentation with an additional text-label class, which is strictly harder; we, therefore, report both the pixel-level metrics and the same figure-level hit rate so that the two formulations remain commensurable. Likewise, the “97%” figure frequently quoted in this literature refers to the label-to-figure association

step rather than to segmentation accuracy, and conflating the two would overstate the state of the art.

Beyond whole figure segmentation, a parallel literature targets the detection and classification of symbols and components inside technical drawings. Elyan et al. applied deep learning to symbol detection and classification in engineering drawings [17], and later work extended this to mechanical engineering sketches with synthetic data generation [18]; reviews of document digitization for complex documents and engineering diagrams consolidate these efforts [19]. What distinguishes our setting from this literature is that none of it models the text label as a pixel-level class inside the segmentation network. Symbols and numerals are treated as separate detection targets, whereas in a compound patent figure the numerals and the sub-figure boundaries are mutually constraining and must be solved together rather than in isolation.

Finally, patent figures have been the object of retrieval, classification and description studies that use large vision-language models [4,5] and of multimodal similarity evaluation [3]. These systems consume figure-level features and, in doing so, implicitly assume that the figure has already been decomposed into its constituent sub-figures and separated from its labels; our work supplies precisely this decomposition as a pre-built, pixel-accurate layer. A distinct and very recent line of work targets compound scientific figures rather than patents, extracting aligned sub-figure-sub-caption pairs from papers with vision-language models [6]. This is further evidence that compound-figure decomposition is an active, high value problem, but it again operates at the granularity of detection and caption association rather than producing a pixel-level segmentation in which sub-figure and text-label regions are jointly modeled under a layout prior.

2.2. Reference Numeral Extraction

Reference numerals and lead lines are the connective tissue of patent drawings, and their extraction has traditionally been treated as a standalone preprocessing problem. The run-length smoothing algorithm (RLSA) [8] merges runs of dark pixels along horizontal and vertical directions so that isolated numeral strokes coalesce into blobs that can subsequently be filtered by geometry; patent-specific text detection applications combine this with optical character recognition and connected component analysis, followed by hand-crafted filters to reject non-text strokes [9,10]. The quality of every downstream step is conditioned on a clean binarization, which is addressed by adaptive thresholds [20] and by learned document enhancement methods [21–23].

More recent benchmarks target the component identifiers themselves at fine granularity, providing detection and OCR annotations for the reference numerals [11]. This is a step toward treating the text as a first-class object, but the extraction still proceeds as an independent, disconnected stage. The structural weakness is the same as in the classical pipelines: these methods do not participate in the optimization of the sub-figure boundaries, so an error in text removal propagates directly into the downstream sub-figure detection. Our formulation instead makes the text-label regions an explicit third semantic class of the segmentation network, so that text separation and sub-figure segmentation are learned jointly and consistently, and a misclassified numeral is penalized by the same objective that optimizes the sub-figure outline.

2.3. Document Layout Analysis

Document layout analysis separates a scanned page into text blocks, figures, tables and other regions, with large-scale benchmarks such as PubLayNet [12] driving supervised approaches. A long companion literature addresses the quality of the input itself, from classic adaptive binarization [20] through learned document enhancement [21,22] to

modern binarization methods [23], all of which improve the reliability of the page-level segmentation that follows.

This body of work, however, operates at the page level: it distinguishes a figure block from a text block but does not descend into the interior of a figure to separate sub-figures from the reference numerals drawn on top of them. That interior granularity is exactly what patent analysis requires, because the informative signal of a compound figure lies in the arrangement of its sub-figures and in the binding of numerals to components. Our work targets precisely this finer level and can be seen as a continuation of layout analysis from the page into the figure itself, replacing the coarse "figure block" with a three-class pixel map of sub-figure, text-label and background regions. The state of the art is also consolidating toward unified document image segmentation: DocSAM, for instance, addresses layout analysis, text segmentation and table recognition within a single query decomposition framework [24]. These methods still treat a figure as one region at the page level; our work is the complementary step of segmenting the interior of that region into sub-figures and their reference labels.

2.4. Vision State-Space Models

State-space models (SSMs) have emerged as an efficient alternative to attention for sequence modeling. The structured state-space model S4 established the line by showing that long-range dependencies can be captured with linear complexity in the sequence length [25]. Mamba advanced this by introducing input-dependent selectivity, so that the state transition depends on the current token and the model learns content-aware forgetting and remembering [26]. The resulting architecture keeps the linear time recurrence of SSMs while regaining the context sensitivity that attention-based models exhibit, which makes it attractive for inputs that are too long or too high-resolution for quadratic attention.

These ideas were subsequently adapted to vision. Vim applied a bidirectional scan over flattened image patches and showed that a pure SSM backbone can compete with vision transformers on ImageNet classification [27]. VMamba proposed the 2-D selective scan (SS2D), which scans the feature map along four cross-scan directions (row-wise, column-wise and the two diagonals) and merges the resulting sequences, thereby restoring the two-dimensional structure that a single raster scan discards and achieving a global receptive field with linear complexity [28]. The SS2D block is the backbone primitive of our method, and its four-direction selective scan is the base onto which we add our two grid directions.

The VMamba family has since been extended along several axes: multi-scale hierarchical variants that strengthen the feature hierarchy [29], ImageNet-based pretraining for medical segmentation [30], semi-supervised training with pixel-level contrastive learning [31], and a rapidly growing body of work surveyed in [32]. For dense prediction, VM-UNet combined the VMamba SS2D backbone with a U-Net decoder, establishing the SS2D-U-Net template that we adopt and extend [33]. The SS2D four-direction scan originates from VMamba [28], while VM-UNet [33] contributes the segmentation head that turns the backbone into a dense predictor.

For our task, the linear-complexity global context of the SS2D backbone is not merely a matter of efficiency. Technical line-art demands high input resolutions to keep fine sub-figure boundaries and the narrow gaps between adjacent panels intact, and an encoder that must attend globally over a large feature map becomes prohibitive under quadratic attention. The selective scan gives the backbone a full image receptive field at a cost that scales linearly with resolution, which is exactly the property needed to segment fine compound figures without destroying their detail through downsampling.

2.5. Scanning Strategies and Direction Design

Because the scan order determines which pixels a selective scan can propagate information between, the design of scanning strategies has become a research topic in its own right. The most systematic study to date is that of Zhu et al., who compared twenty-two scanning strategies for semantic segmentation of remote sensing imagery and provided a controlled reference for direction design [34]. Their findings confirm that the choice of scan directions materially affects accuracy, while also showing that no single generic ordering dominates across all inputs.

Subsequent work has explored strategies beyond fixed raster and diagonal orderings. Switch-UMamba introduces a dynamic scanning mechanism that selects among scan patterns based on the input, rather than committing to a fixed set [35]. CrackSnakeNet uses a multi-directional snake scanning pattern designed to preserve the topology of thin, elongated cracks [36]. Symmetrical scanning has been proposed for image deraining [37], and a series of domain-oriented designs have targeted remote sensing, including RS-Mamba for large remote sensing images [38] and link aggregation strategies for skip connections [39]. Closer to the medical and fine structure regime, hierarchical multi-scale scanning with tubular structure-aware convolution has been applied to retinal vessel segmentation [40].

A common property of these strategies is that they are task-agnostic: they either expand the set of directions generically, adapt the scan dynamically from the input, or impose a smoothness prior suited to their own domain, but none of them encodes the layout structure of the target content. In particular, to the best of our knowledge, none of the twenty-two strategies studied by Zhu et al. [34] encodes a grid layout prior, and no prior work exploits the fact that a compound patent figure is organized as a grid of sub-figures. Our grid-aware scan is the first to derive additional scan directions from this domain structure: it appends row-priority and column-priority traversals over a grid partition, so that the selective scan follows the figure layout rather than only the raster lines, and it does so without the added complexity of dynamic or topology-specific scan machinery.

2.6. Positioning

In summary, prior work either performs binary sub-figure segmentation that discards the text, extracts reference numerals as a disconnected heuristic, or analyzes documents at the page level without entering the figure and existing vision-Mamba scanning strategies, however sophisticated, are task-agnostic with respect to layout. PatMamba is, to our knowledge, the first framework that (i) predicts the text labels as a pixel-level semantic class jointly with the sub-figure regions and (ii) encodes the compound-figure grid structure explicitly into the selective-scan directions of a Vision Mamba backbone, thereby occupying the unclaimed intersection of technical drawing segmentation and structure-aware state-space modeling.

3. Method

3.1. Problem Formulation

Patent documents express much of their technical content in drawings: a single compound figure typically packs several sub-figures into one image and overlays them with a dense mesh of reference numerals, letters and lead lines that annotate individual parts. Downstream tasks such as prior art search, infringement comparison and figure-text association all begin by decomposing a compound figure into its constituent sub-figures and separating the annotation text from the line-art. We, therefore, formulate the task as a dense three-class semantic segmentation, rather than the binary "figure vs. background" detection used in earlier work: class 0 is the background, class 1 the sub-figure regions, and

class 2 the text-label regions. Making the text labels an explicit semantic class rather than treating them as noise to be removed afterward is the first design choice that distinguishes our formulation, because the labels sit directly on and across the sub-figure boundaries and, if left unmodeled, contaminate both the segmentation and any subsequent sub-figure detection step.

Formally, let a compound patent figure be a grayscale image $\mathbf{X} \in \mathbb{R}^{H \times W}$ (white background, black line-art) and let $\mathbf{Y} \in \{0,1,2\}^{H \times W}$ be its dense label map. The ground truth is obtained by rasterizing the DeepPatent2 annotations: every sub-figure bounding box $(x, y, w_{\text{figure}})$ is filled as class 1, every normalized text-label box (W, H, L, T) as class 2, and every remaining pixel as class 0. The boxes are rasterized onto the pixel grid before training, so the labels are a deterministic function of the annotation file and the resize operation; no manual refinement is applied. This ground truth defines the task at the granularity of bounding-box localization rather than stroke-level boundary extraction: the label of a sub-figure is its annotation rectangle, not the exact ink boundary of the drawing, because the public DeepPatent2 annotations provide boxes rather than pixel masks. The task is, therefore, best understood as "where are the sub-figures and their labels", and the pixel-level metrics in Section 4.2.2 are interpreted as box fitting fidelity; we additionally report a boundary-band IoU to quantify how tightly a predicted boundary hugs the annotated box. PatMamba learns a mapping $F_\theta : \mathbf{X} \mapsto \mathbf{P}$, where

$$\mathbf{P} = \text{softmax}\big(g_\phi(f_\theta(\mathbf{X}))\big) \in \mathbb{R}^{3 \times H \times W}, \tag{1}$$

f_θ is the SS2D backbone and g_ϕ the U-Net decoder; the prediction is the per-pixel argmax $\hat{\mathbf{Y}}_{ij} = \arg\max_c \mathbf{P}_{c,ij}$. A single end-to-end forward pass thus yields, simultaneously, the sub-figure mask (argmax of class 1) and the text-label mask (argmax of class 2), which is exactly the input required by the downstream decomposition and detection pipeline. The notation used throughout the paper is summarized in Table 1.

Table 1. Nomenclature.

Symbol	Meaning
$\mathbf{X} \in \mathbb{R}^{H \times W}$	input grayscale compound patent figure
$\mathbf{Y} \in \{0,1,2\}^{H \times W}$	dense label map (0 background/1 sub-figure/2 text-label)
$\mathbf{P} \in \mathbb{R}^{3 \times H \times W}$	per-pixel class probability (softmax output)
f_θ, g_ϕ	SS2D backbone/U-Net decoder
z	normalized feature map entering an SS2D block
π_k, D	k-th scan permutation; number of directions (4 or 6)
A, B, C, Δ	SSM state transition/input/output/step (input-selective)
N	SSM state dimension ($d_{\text{state}} = 16$)
$\mathcal{C}_{ij}$, g_h, g_w	grid cell (i, j); grid size (2×2)
$\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{dice}}$	cross-entropy/soft Dice loss

3.2. Overall Architecture

Figure 1 gives an overview of PatMamba. The design follows the VMamba SS2D backbone + U-Net decoder template, a combination that is well matched to the two requirements of technical drawing segmentation: the SS2D backbone provides a global receptive field at linear cost in the number of pixels, which is essential at the high resolutions needed to resolve the thin boundaries between adjacent sub-figures, while the U-Net decoder restores the fine boundaries that downsampling inevitably blurs. The architecture is kept deliberately symmetric and simple, because the target signal, thin black lines on a white canvas, is sparse and boundary-sensitive: a backbone that is too deep and heavily downsampled would erase the finest strokes, whereas one that is too shallow cannot observe a whole multi-panel figure at once.

The data flow proceeds as follows. The grayscale input is first projected by a patch-embedding stem to a 1/4-resolution feature map, then processed by four SS2D stages of increasing channel width and decreasing resolution, yielding features at 1/4, 1/8, 1/16 and 1/32 of the input scale. Each stage deepens the receptive field and abstracts the representation, while the proposed grid-aware scan, which is present at every stage and hence operates at multiple scales, keeps the compound-figure layout structure in view throughout. The decoder then upsamples these features and fuses them through skip connections back to full resolution, and a 1×1 convolution produces the three-class segmentation. The core contribution, the grid-aware scan, replaces the generic four-direction cross-scan inside each SS2D block with a six-direction scan that appends two grid layout directions, injecting the layout structure into the selective-scan sequence modeling without any architectural change elsewhere. Table 2 summarizes the stage configuration.

Table 2. Backbone stage configuration of the Tiny (VMamba-T) scale used in PatMamba.

Stage	Resolution	Channels	Blocks
Patch embedding	1/4	96	—
Stage 1	1/4	96	2
Stage 2	1/8	192	2
Stage 3	1/16	384	6
Stage 4	1/32	768	2

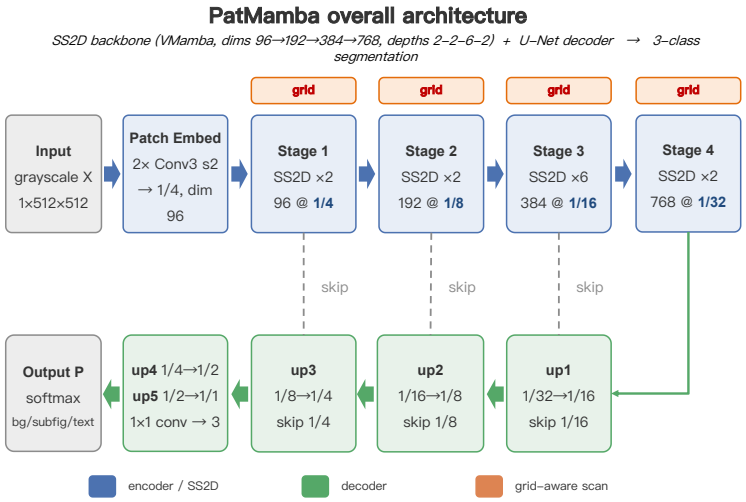


Figure 1. Overview of the proposed PatMamba. A grayscale compound patent figure is encoded by a VMamba SS2D backbone into four multi-scale features (1/4, 1/8, 1/16, 1/32); every SS2D block is equipped with the proposed grid-aware scan (orange: grid-aware scan); gray boxes (gray: input/output modules) denote input and output modules. A U-Net decoder with skip connections upsamples the features to full resolution and a 1×1 convolution produces the 3-class segmentation.

3.3. SS2D Backbone

3.3.1. Patch Embedding and Stages

The stem consists of two overlapping 3×3 convolutions of stride 2, which reduce the input to 1/4 resolution with an embedding dimension of 96, following VMamba. Using two overlapping stride-2 convolutions rather than a single non-overlapping patch projection gives the stem a small receptive field, so that fine boundary detail is aggregated before any downsampling; this matters for line-art, where a one-pixel shift of a sub-figure boundary can already merge two adjacent panels. The backbone then has four stages with channels $\{96, 192, 384, 768\}$ and block depths $\{2, 2, 6, 2\}$, separated by a 3×3 stride-2 convolution that halves the resolution and doubles the channels. The depth is concentrated in the third (middle) stage and the channel budget grows toward the coarsest

stage, matching the observation that the discriminative structure of a technical drawing lives at intermediate resolutions. The whole backbone (the Tiny scale, about 22M parameters) yields features at 1/4, 1/8, 1/16 and 1/32 resolution, which the decoder later recombines.

3.3.2. Cross-Scan (Four Directions)

State-space sequence models operate on a 1-D sequence, whereas a figure is a 2-D signal, so the feature map must be flattened into a sequence. The order in which pixels are visited matters deeply, because the S6 recurrence only propagates state forward along the sequence: a naive row-major flattening of $z \in \mathbb{R}^{C\times H\times W}$ into a sequence of length $L = HW$ would let each pixel see only the pixels above and to its left, leaving a strong directional bias. The cross-scan removes this bias by scanning in several complementary orders. Each order is a permutation π_k of the index set $\{0, \ldots, L-1\}$, and after the scan the inverse permutation π_k^{-1} writes the outputs back to their original 2-D positions. The four cross-scan orders are as follows:

$$\pi_0(i,j) = iW + j, \quad \pi_1 = \mathrm{rev}(\pi_0), \quad \pi_2(i,j) = jH + i, \quad \pi_3 = \mathrm{rev}(\pi_2), \tag{2}$$

that is, the two opposite row directions and the two opposite column directions (Figure 2). Because a given pixel appears at a different position in each of the four sequences, its context after the merge comes from all four sides of the image; in particular, the opposite directions pair up so that every position can aggregate context from both ends of each axis. This four-direction mechanism is the key property inherited from VMamba: it gives a global receptive field at every layer while keeping the computation linear in L.

3.3.3. Selective Scan (S6)

Each permuted sequence is passed to a selective state-space (S6) layer, the recurrent core of the Mamba family. The layer maintains a small hidden state $h \in \mathbb{R}^N$ that is updated one token at a time and acts as a running summary of the sequence seen so far: content deemed relevant is written into the state and carried forward over many positions, while irrelevant content decays. Along a permuted 1-D sequence $u = (u_1, \ldots, u_L)$ of length L, the recurrence with state dimension N reads as follows:

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t u_t, \qquad y_t = C_t h_t + D\, u_t, \tag{3}$$

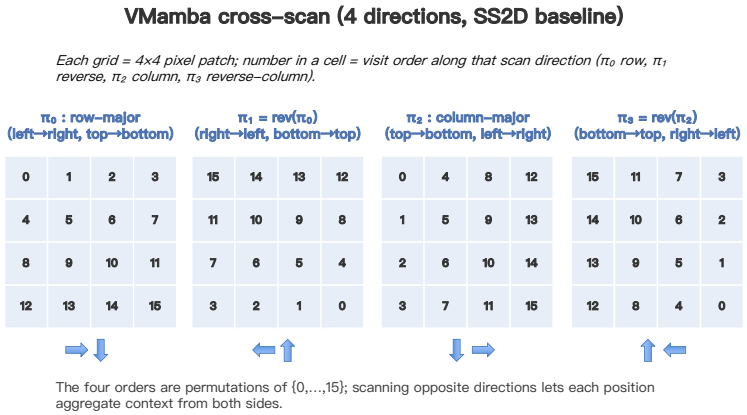
Unlike a classical linear state-space model, whose matrices are fixed after training, the S6 layer makes its parameters input-dependent; this is the selectivity that distinguishes Mamba from linear time-invariant state-space models:

$$\Delta_t = \mathrm{softplus}\big(s_\Delta(u_t)\big), \quad B_t = s_B(u_t), \quad C_t = s_C(u_t), \tag{4}$$

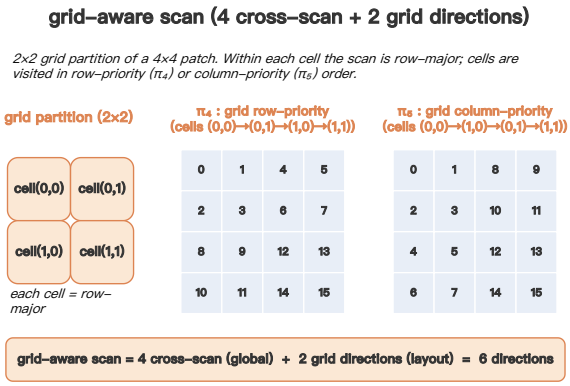
$$\bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t \approx \Delta_t B_t, \tag{5}$$

where $A \in \mathbb{R}^{N\times N}$ is the state transition matrix and the zero-order-hold rule discretizes the continuous dynamics exactly for piecewise-constant inputs. Intuitively, the input-dependent step Δ_t acts as a learnable forget/remember gate: a small Δ_t leaves the state almost unchanged (the model skips or forgets), while a large Δ_t lets the current token overwrite the state. This is precisely the behavior needed for technical line-art, where the same semantic object (a sub-figure outline or a long lead line) is frequently interrupted by background gaps yet must be tracked as one connected structure. The state dimension is fixed at $N = d_{\text{state}} = 16$, small enough to keep each recurrence cheap while still summarizing long-range context. Before entering the scan, the features are prepared by a 1-D depthwise convolution of kernel 4 (which injects local positional structure into the sequence) and by

input/output projections with expansion factor expand $= 2$, doubling the width in the intermediate projections so that the selective parameters have sufficient capacity.



(**a**) Cross-scan (4 directions).



(**b**) Grid-aware scan (6 directions).

Figure 2. (**a**) The four cross-scan directions of the VMamba SS2D block. (**b**) The proposed grid-aware scan: the four cross-scan directions are retained, and two grid directions (row-priority and column-priority over a $g_h \times g_w$ partition) are appended, so the selective scan explicitly follows the compound-figure layout.

3.3.4. SS2D Block

A full SS2D block wraps the scan in a residual, gated structure (Figure 3). The input is first normalized, then split into the D scan directions; each direction is permuted, processed by its own S6 recurrence, and written back to the 2-D grid; the D outputs are averaged into a single feature map, gated by an input-dependent sigmoid mask, and added back through a residual connection:

$$z = \mathrm{LN}(x), \qquad m = \frac{1}{D}\sum_{k=0}^{D-1} \pi_k^{-1}\Big(\mathrm{S6}\big(\pi_k(z)\big)\Big), \qquad \mathrm{out} = x + \sigma(W_g\, z) \odot m, \tag{6}$$

where π_k^{-1} is the inverse permutation that writes each direction's output back to its 2-D positions, $D = 4$ for the baseline and $D = 6$ for the grid-aware scan (Section 3.4), $\odot$ is the element-wise product and σ is the sigmoid gate. First, averaging rather than summing across directions keeps the output magnitude independent of D, so the 4- and 6-direction variants remain directly comparable without retuning any learning rate or normalization; this is what makes the grid-aware scan a clean, controlled insertion. Second, all directions share the SSM parameters (the transition matrix and the input/output projections), except for a small per-direction projection for Δ; the directions differ only in the order in which they visit pixels, not in the transformation they apply. The residual connection and the

sigmoid gate follow the standard gated pattern and ensure that the scan refines the input rather than recomputing it from scratch.

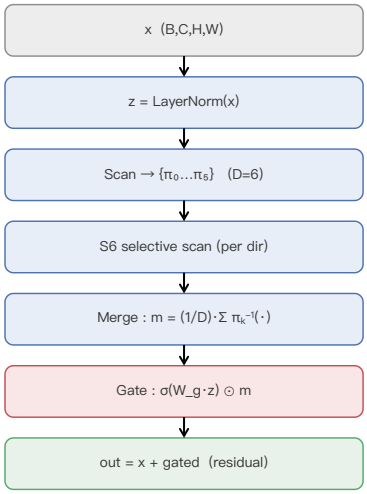


Figure 3. Internals of the SS2D block (gray: input module; blue: main computation units; red: gate unit; green: residual output). LayerNorm is followed by the multi-direction scan, the per-direction selective scan (S6 recurrence on the right), the averaging merge, the sigmoid gate and the residual connection; the grid-aware variant uses $D = 6$ scan orders.

3.4. Grid-Aware Scan

3.4.1. Motivation

A compound patent figure is not an arbitrary natural scene but a grid of sub-figures arranged in rows and columns (2×2, 1×4, 3×2, and irregular tilings). The generic cross-scan traverses the whole image along straight raster lines, so a 1-D selective scan crosses sub-figure boundaries as if they were ordinary pixels: the recurrent state does not know that a row entering panel $(1, 2)$ from panel $(1, 1)$ has crossed a meaningful layout boundary, and it cannot align its context with the row/column structure that the human annotator used. In our data, 71.2% of figures contain a single sub-figure, yet the hard and practically important cases are precisely the dense multi-panel figures (four or more sub-figures), where sub-figures are small, boundaries are thin and detection accuracy is lowest. The grid-aware scan addresses exactly this case. Its design goal is deliberately minimal: to make the scan layout-aware by changing only the order in which pixels are visited, without altering the backbone, the block, or the parameter count, so that any improvement can be attributed cleanly to the injected layout prior. Concretely, it appends two grid layout scan directions to the four cross-scan directions.

3.4.2. Grid Partitioning

The $H \times W$ plane is partitioned into a $g_h \times g_w$ grid of rectangular cells. Rectangular cells, rather than arbitrary shapes, are chosen because patent sub-figures are axis-aligned rectangles, so a rectangular tiling is the correct inductive bias and requires no learned shape model. Cell (i, j) covers the pixel set

$$\mathcal{C}_{ij} = \{(r, c) : r \in [r_i, r_{i+1}),\ c \in [c_j, c_{j+1})\}, \qquad r_i = \text{round}\left(\tfrac{iH}{g_h}\right),\ c_j = \text{round}\left(\tfrac{jW}{g_w}\right), \quad (7)$$

with the last row and column absorbing the rounding remainder, so the cells tile the image exactly even when H or W is not divisible by g_h or g_w. This integer index formulation makes the scan order a pure permutation of pixel indices; it requires no resampling or interpolation, adds no parameters, and is trivially invertible. By default $g_h = g_w = 2$, a

value that balances the layout prior against the size of the cells; the sensitivity to this choice is studied in the ablations of Section 4.

3.4.3. Augmented Six-Direction Scan

The grid-aware scan keeps the four whole-image cross-scan directions, thereby preserving the global receptive field because those directions still traverse the entire image, and appends two grid directions in which pixels are visited cell by cell (Figure 2):

- π_4 (grid row-priority): cells in row-major order $(i = 0, \ldots, g_h - 1;\ j = 0, \ldots, g_w - 1)$, pixels inside each cell in row-major order;
- π_5 (grid column-priority): cells in column-major order $(j = 0, \ldots, g_w - 1;\ i = 0, \ldots, g_h - 1)$.

The resulting sequence first traverses within each sub-figure-like cell, so the state saturates on one panel before moving on, and then moves across cells in reading order, so the state recurrently links neighboring sub-figures along the layout rather than only along raster lines. Inside a cell, the model aggregates the local line-art, while at each cell boundary it transitions explicitly to the next panel, a structure that the generic cross-scan can only approximate implicitly. The two appended directions mirror the bidirectional logic of the cross-scan: row-priority links cells left-to-right and top-to-bottom, column-priority links top-to-bottom and left-to-right, so that a panel's context comes from both its horizontal and its vertical neighbors. Formally,

$$\pi_4 = \underset{(i,j)}{\text{concat}}\ \mathcal{C}_{ij}, \qquad \pi_5 = \underset{(j,i)}{\text{concat}}\ \mathcal{C}_{ij}. \tag{8}$$

The grid-aware SS2D block is exactly Equation (6) with $D = 6$ and scan orders $\{\pi_0, \ldots, \pi_5\}$; the merge still averages over all six directions, and the parameter count is essentially unchanged (analyzed in the complexity paragraph below). Because the grid directions are only permutations, they introduce no new operator: the very same S6 recurrence is reused for all six directions, which keeps the modification a drop-in change with a negligible implementation footprint. The index construction is summarized in Table 3 and the block forward pass is in Table 4.

Table 3. Grid-augmented scan index generation.

Input:	height h, width w, grid size $g_h \times g_w$
Output:	six permutations $\{\pi_0, \ldots, \pi_5\}$ of $\{0, \ldots, hw - 1\}$
1	$\{\pi_0, \pi_1, \pi_2, \pi_3\} \leftarrow \text{CrossScan}(h, w)$ Equation (2)
2	**for** $i = 0$ **to** $g_h - 1$, $j = 0$ **to** $g_w - 1$ **do**
3	$\quad \mathcal{C}[i, j] \leftarrow$ pixels of cell (i, j) in row-major order
4	$\pi_4 \leftarrow \text{concat}_{(i,j)}\, \mathcal{C}[i, j]$ row-priority cells
5	$\pi_5 \leftarrow \text{concat}_{(j,i)}\, \mathcal{C}[i, j]$ column-priority cells
6	**return** $\{\pi_0, \ldots, \pi_5\}$

Table 4. SS2D block with grid-aware scan (forward).

Input:	feature $x \in \mathbb{R}^{B \times C \times H \times W}$, scan module S, selective-scan S6
Output:	feature $y \in \mathbb{R}^{B \times C \times H \times W}$
1	$z \leftarrow \text{LayerNorm}(x)$
2	$m \leftarrow 0$
3	**for** $k = 0$ **to** $D - 1$ **do** $D = 4$ baseline; $D = 6$ grid-aware scan
4	$\quad s \leftarrow S.\text{scan}(z, \pi_k)$ permute to (B, C, L)
5	$\quad h \leftarrow \text{S6}(s)$ Equations (3)–(5)
6	$\quad m \leftarrow m + \pi_k^{-1}(h)$ write back to 2-D
7	$m \leftarrow m / D$
8	$g \leftarrow \sigma(W_g z)$
9	**return** $x + g \odot m$

3.4.4. Complexity and Parameter Cost

All SSM parameters are shared across the D directions; only the per-direction Δ-projection s_Δ is independent. Going from $D = 4$ to $D = 6$, therefore, adds exactly two extra per-direction Δ-projections per SS2D block, each a $d_{\text{rank}} \times 2\,\text{dim}$ linear map with $d_{\text{rank}} = \text{dim}/16$. Summed over the twelve blocks of the Tiny backbone (dimensions 96/192/384/768 at depths 2/2/6/2), this amounts to $\approx$0.56 M parameters in total, a $+$2.0% increase over the 27.96 M parameters of the full VM-UNet model and a negligible parameter overhead against the backbone it augments. The scan and merge steps are pure index permutations with $O(HW)$ cost and no learned parameters, contributing no matrix multiplications to the block. The dominant cost of the SS2D block remains the S6 recurrence, which is $O(L \cdot N \cdot \text{expand})$ per direction and hence linear in the number of pixels $L = HW$. The grid-aware scan, therefore, preserves VMamba's linear complexity and near-zero parameter overhead, with only a constant factor increase in the number of scan passes from four to six. In practice, the permutation is a memory bound gather/scatter whose wall clock cost is small relative to the recurrence; we verify this empirically in the efficiency analysis of Section 4.

3.5. U-Net Decoder and Segmentation Head

The decoder restores full resolution from the four backbone features $\{f_{1/4}, f_{1/8}, f_{1/16}, f_{1/32}\}$ through a cascade of five decoder blocks. Each block (i) upsamples its input by a factor of 2 with bilinear interpolation, (ii) concatenates the same scale backbone feature when one exists (the skip connection), and (iii) applies two 3×3 convolutions with SiLU activation. Bilinear upsampling is preferred over transposed convolutions because it introduces no learned upsampling kernel and hence no checkerboard artifacts, which would be especially visible on the thin boundaries of technical line-art. The channel widths follow $768 \to 384 \to 192 \to 96 \to 48 \to 48$, with skip connections fusing the $1/16$, $1/8$ and $1/4$ backbone features; a final 1×1 convolution maps the 48-channel feature to the three-class logits:

$$\mathbf{P} = \text{softmax}\left(\text{Conv}_{1\times1}\left(\text{Dec}(\{f_{1/4}, f_{1/8}, f_{1/16}, f_{1/32}\})\right)\right). \tag{9}$$

A U-Net decoder, rather than a lightweight feature pyramid head, is chosen deliberately: the segmentation targets, namely sub-figure boundaries and text labels, are dominated by fine, low-level edges, and the skip connections reinject exactly the low-level detail that four stages of downsampling have abstracted away. The two 3×3 convolutions per block let each scale reconcile the coarse semantic context propagated from the decoder with the fine spatial detail from the skip connection before the next upsampling, which prevents boundary blur at the seams between scales.

3.6. Loss Function

PatMamba is trained with a combination of two complementary losses: the cross-entropy loss and the soft Dice loss. The combination is motivated by the severe class imbalance of patent figures, in which the background occupies the large majority of pixels while the sub-figure and text-label classes are thin and sparse; consequently, a per-pixel classification loss alone would be dominated by the easy background class. The cross-entropy term supplies a smooth, well-behaved gradient for per-pixel classification, while the Dice term optimizes the region overlap directly and is robust to the number of background pixels by construction:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\sum_{c} y_{ic}\log p_{ic}, \tag{10}$$

$$\mathcal{L}_{\text{dice}} = 1 - \frac{1}{C}\sum_{c}\frac{2\sum_i p_{ic}\, y_{ic} + \varepsilon}{\sum_i p_{ic} + \sum_i y_{ic} + \varepsilon}, \tag{11}$$

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda\, \mathcal{L}_{\text{dice}}, \qquad \lambda = 1.0, \tag{12}$$

where p_{ic} is the predicted probability of class c at pixel i, $y_{ic} \in \{0, 1\}$ is the one-hot label, $C = 3$ is the number of classes and $\varepsilon = 1.0$ is a smoothing constant that keeps the Dice term well-defined when a class is absent from a crop. The two terms are combined with equal weight $\lambda = 1.0$: the cross-entropy term stabilizes early training and keeps the logits calibrated, while the Dice term directly rewards the exact region overlap that the evaluation metrics measure. A boundary-aware auxiliary loss [41] was also evaluated as a candidate refinement but produced no measurable gain over this combination and was, therefore, not retained.

For the content (sub-figure) class, we further apply a content-recall weight w: the class mean $\frac{1}{C}\sum_c$ in Equation (11) is replaced by the class weighted mean $\sum_c w_c\, \text{Dice}_c \,/\, \sum_c w_c$ with $w_c = [1.0,\, w,\, 1.0]$ for the background, sub-figure and text-label classes. The weight counteracts an under-segmentation of the content class under distribution shift; it is in-domain insensitive (Section 4.4.4), and we set $w = 1.8$ for the cross-domain study (Section 4.7).

4. Experiments and Results

4.1. Dataset and Ground-Truth Construction

We use the 2010 subset of DeepPatent2 [2], a large-scale benchmark of US patent technical drawings. Each compound figure is accompanied by an annotation file that lists, for every sub-figure, its bounding box $(x, y, w_{\text{figure}})$, and, for every text label, its normalized bounding box (W, H, L, T). We rasterize these boxes onto the pixel grid to obtain the dense three-class label map described in Section 3.1: sub-figure boxes are filled as class 1, text-label boxes as class 2, and all remaining pixels as class 0. Table 5 summarizes the dataset statistics. Consistent with the known distribution of patent drawings, the large majority of figures contain a single sub-figure (71.2%), while only a small fraction (2.2%) are regular 2×2 grids; the dense multi-panel figures are consequently the minority and the most challenging cases.

Table 5. Dataset statistics of the DeepPatent2 (2010) subset.

Statistic	Value
Compound figures (total)	80,553
Sub-figure bounding boxes (total)	112,164
Text-label bounding boxes (total)	112,164
Training/validation/test figures	48,358/15,882/16,313
Figures with a single sub-figure	71.2%
Figures with a regular 2×2 grid	2.2%

Note: the sub-figure and text-label totals coincide because the DeepPatent2 annotation stores exactly one text label per sub-figure, i.e., one record per sub-figure together with its label, so both counts equal the number of records.

In addition to this in-distribution benchmark, we evaluate zero-shot transfer to four external corpora (utility patents, biomedical compound figures, chemical reaction schemes and handwritten circuit diagrams), which are introduced together with their annotation formats in Section 4.7.

4.2. Experimental Setup

4.2.1. Split Protocol

We split the dataset at the patent level into 60/20/20 training/validation/test partitions using a fixed random seed (SEED = 42). Because all figures of a patent are assigned to the same split, no figure of a given patent appears in more than one partition, which prevents leakage of near-duplicate drawings across the training and evaluation sets. The validation partition contains 15,882 figures and is used for all results reported in this section. The held-out test partition contains 16,313 figures and was evaluated exactly once, after model selection was finalized, so that no hyperparameter decision was informed by the test set. Table 6 reports the complete test-set results; they closely reproduce the validation numbers (grid-aware scan mIoU 0.9174 vs. 0.9178), confirming that the reported numbers generalize to unseen patents rather than overfitting the validation partition.

Table 6. Held-out test-set results (pixel-level metrics).

Model	mIoU	mDice	BF1	IoU_0	IoU_1	IoU_2
U-Net	0.8393	0.9125	0.5481	0.8088	0.8502	0.8588
U-Net† (matched budget)	0.8399	0.9129	0.5524	0.8119	0.8508	0.8570
SegFormer (MiT-B0)	0.9058	0.9500	0.6287	0.9253	0.9468	0.8454
VM-UNet (baseline)	0.9169	0.9562	0.7602	0.9353	0.9541	0.8613
PatMamba (grid 2×2)	0.9174	0.9564	0.7671	0.9365	0.9549	0.8608

4.2.2. Evaluation Metrics

We evaluate with two complementary families of metrics. At the pixel level, we report the mean Intersection over Union (mIoU), the mean Dice coefficient (mDice), the Boundary F1 score (BF1), together with the per-class IoU of the three classes (background IoU_0, sub-figure IoU_1, text-label IoU_2). To directly assess boundary fidelity under the ground truth rasterized from boxes, we additionally report the boundary-band IoU (BB-IoU_d): the foreground IoU restricted to the band of $\pm d$ pixels around the ground-truth sub-figure boundary, for $d \in \{3, 5\}$. A high BB-IoU_d means the predicted boundary lies within d pixels of the annotated box, isolating the localization quality that a plain box filling mIoU cannot separate from interior fill accuracy. At the figure level, following the protocol of prior patent figure segmentation work [7], we report the sub-figure detection hit rate: a predicted sub-figure is counted as a hit if its bounding-box IoU with a ground-truth sub-figure exceeds a threshold (0.7 or 0.9), which aligns our numbers with the MedT-style metric.

4.2.3. Implementation Details

All models are trained at an input resolution of 512×512 with a batch size of 8 using the AdamW optimizer with a learning rate of 5×10^{-4}, a fixed random seed of 42 for both data shuffling and weight initialization, and full fp32 precision (the mamba-ssm kernels available in our environment do not support fp16). Training is performed on a single NVIDIA RTX 4090 GPU. We report the optimization budget in iterations, which are invariant to the dataset and batch sizes. The proposed PatMamba and its VM-UNet backbone baseline are trained for $\approx 9 \times 10^4$ iterations (15 epochs over the ≈48K-figure training split at batch size 8), while the external baselines (U-Net, SegFormer) follow their standard 25-epoch schedule. To separate the effect of the training budget from the backbone itself, we additionally train a budget-matched U-Net† with the same $\approx 9 \times 10^4$-iteration budget as the Mamba models for fair comparison against the standard U-Net. The iteration budget was chosen after pilot experiments showed that the validation metrics plateau around 7×10^4–9×10^4 iterations (12–15 epochs), with no additional gain from longer schedules. Table 7 lists the key hyperparameters.

4.3. Main Results

The comparison spans the four method families that define the pixel-level segmentation task at hand: a traditional heuristic (RLSA + connected components), a CNN (U-Net and its budget matched variant U-Net†), a Transformer (SegFormer) and the Mamba backbone (VM-UNet). We exclude contrastive vision-language encoders such as CLIP [42]: they emit image- or region-level embeddings for retrieval and open-vocabulary association rather than a per-pixel, three-class segmentation mask, and adapting one to our task would require attaching a segmentation decoder to a frozen encoder, which reduces the comparison to "which decoder on which encoder" rather than the backbone comparison performed in this section.

Table 7. Training hyperparameters.

Hyperparameter	**Value**
Optimizer/learning rate	AdamW/5×10^{-4}
Iterations (epochs)	$\approx 9 \times 10^4$ (15)
Batch size	8
Input resolution	512×512
Weight initialization	random (SEED = 42)
Precision	fp32
GPU	NVIDIA RTX 4090 24 GB

4.3.1. Sub-Figure Detection

In the in-domain tables, PatMamba is reported as the grid-aware scan without the content-recall weighting; the weighting is in-distribution neutral (Section 4.3.3) and is reserved for the cross-domain study (Section 4.7). Table 8 reports the figure-level sub-figure detection hit rate. The transition from the traditional run-length-smoothing baseline [8] to learned methods improves the hit rate by +23.7 points at $\mathrm{IoU}_{0.7}$ and by +40.6 points at $\mathrm{IoU}_{0.9}$. The ranking across backbones is clear: Mamba (≈94.5%) outperforms the Transformer SegFormer [43] (93.7%) and, by a large margin, the CNN U-Net [15] (72.6%). Notably, the CNN's hit@$\mathrm{IoU}_{0.9}$ (43.3%) is lower than the traditional baseline (50.1%): although U-Net attains a respectable pixel mIoU (0.842), its sub-figure boundaries are blurred and neighboring sub-figures tend to merge, so it cannot extract clean sub-figure boxes. Training U-Net with the same $\approx 9 \times 10^4$-iteration budget as the Mamba models (U-Net†, Table 8) recovers only a small fraction of this gap (76.90%/45.44%), confirming that the CNN deficit is architectural rather than a training budget artifact. This contrast shows that figure-level detection and pixel-level mIoU are two distinct metrics, and that the Mamba backbone's advantage on clean sub-figure boundaries surfaces most clearly at the figure level, where blurred boundaries merge adjacent panels and degrade the box-IoU hit rate. The grid-aware scan leaves the detection hit rate essentially unchanged (94.39%/90.85% versus 94.47%/90.75%), as expected, since its contribution is localized to the segmentation of dense multi-panel figures rather than to whole figure localization. Figure 4 visualizes this ladder.

Table 8. Figure-level sub-figure detection (hit rate at box-IoU $\geq$ 0.7 and $\geq$0.9).

Method	Paradigm	hit@IoU$_{0.7}$	hit@IoU$_{0.9}$
RLSA + connected components	Traditional	70.74%	50.11%
U-Net	CNN	72.56%	43.29%
U-Net† (matched budget)	CNN	76.90%	45.44%
SegFormer (MiT-B0)	Transformer	93.67%	88.19%
VM-UNet (baseline)	Mamba	94.47%	90.75%
PatMamba (grid 2 $\times$ 2)	Mamba + grid	94.39%	90.85%

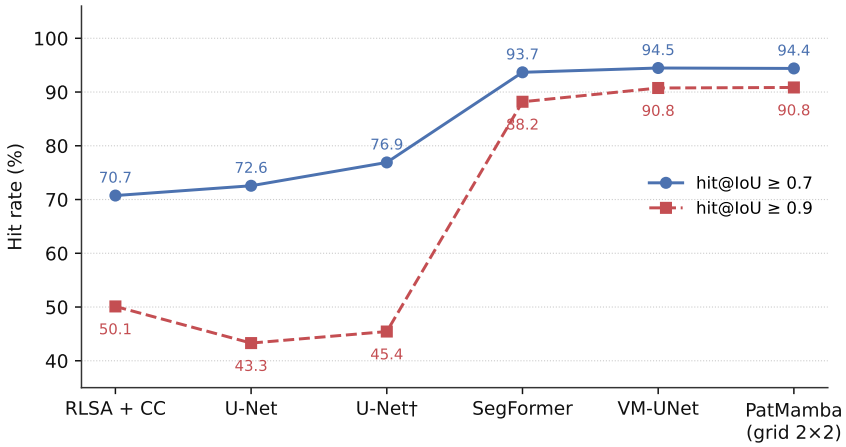


Figure 4. Figure-level sub-figure detection hit rate across methods at box-IoU thresholds of 0.7 and 0.9. The traditional run-length-smoothing baseline and the CNN fall far below the Transformer and Mamba models; note that U-Net's hit@IoU$_{0.9}$ drops below the traditional baseline, reflecting its blurred sub-figure boundaries. The budget matched U-Net† recovers only a fraction of this gap, confirming that the deficit is architectural rather than a training budget artifact.

4.3.2. Pixel-Level Segmentation

Table 9 reports the pixel-level segmentation results. The Mamba models dominate across all metrics: VM-UNet improves the mIoU from 0.842 (U-Net) and 0.906 (SegFormer) to 0.916, and the Boundary F1 from 0.550/0.631 to 0.761. The budget matched U-Net† attains 0.8431 mIoU, only 0.0013 above U-Net, which confirms that the Mamba advantage is not an effect of the training budget. The grid-aware scan improves every metric further, with the relative gain largest on the Boundary F1 (+0.7 points), consistent with its effect on sub-figure boundaries in dense layouts. Under the ground truth rasterized from boxes, as defined in Section 3.1, both Boundary F1 and the boundary-band IoU (BB-IoU) measure alignment with the annotated box, so this boundary gain should be read as box boundary fidelity rather than stroke-level precision. The text-label class (IoU$_2$) remains the hardest class for the Mamba and Transformer backbones, reflecting the fine and sparse nature of the numerals; for the CNN it is instead the background class, another consequence of its blurred sub-figure boundaries.

Table 9. Pixel-level semantic segmentation (validation set).

Model	mIoU	mDice	BF1	IoU$_0$	IoU$_1$	IoU$_2$
U-Net	0.8418	0.9140	0.5498	0.8149	0.8527	0.8578
U-Net† (matched budget)	0.8431	0.9148	0.5559	0.8185	0.8537	0.8571
SegFormer (MiT-B0)	0.9059	0.9500	0.6308	0.9275	0.9472	0.8429
VM-UNet (baseline)	0.9163	0.9558	0.7606	0.9369	0.9543	0.8576
PatMamba (grid 2 $\times$ 2)	0.9178	0.9566	0.7680	0.9383	0.9553	0.8598

4.3.3. Statistical Significance

The headline gains are small on the aggregate (the grid 2×2 scan improves the mIoU over the baseline by only $+0.15$ points, Table 10); we, therefore, assess robustness across five random seeds and isolate the contribution of the content-recall weighting. We train three model groups, each across five seeds $\{42, 123, 2024, 2025, 2026\}$: the proposed PatMamba (content-recall weighting $w = 1.8$), the VM-UNet baseline, and an unweighted control (the "PatMamba (grid 2×2)" of Tables 8 and 9) that differs from PatMamba only by removing the class weights. We report paired t-tests (df $= 4$), Wilcoxon signed-rank tests, pooled Cohen's d and bootstrap 95% confidence intervals over the seeds. Table 11 decomposes the total effect of PatMamba versus the baseline into an architecture component (the grid-aware scan, control versus baseline) and a weighting component (PatMamba versus control), evaluated on the dense $\geq$ 5-panel bucket where the effect of Section 3.4 is concentrated (n= 137).

Table 10. Gain localization of the grid-aware scan, bucketed by the number of sub-figures (grid 2×2 versus baseline).

# Sub-Figures	# Images	Base mIoU	Grid mIoU	Base IoU_1	Grid IoU_1
All	15,882	0.9163	0.9178	0.9543	0.9553
1	11,119	0.9180	0.9196	0.9599	0.9607
$\geq$2	4763	0.9111	0.9127	0.9375	0.9393
2	3726	0.9216	0.9228	0.9485	0.9499
3	627	0.8779	0.8771	0.8987	0.8949
4	273	0.8765	0.8867	0.8884	0.9015
$\geq$5	137	0.8743	0.8806	0.8783	0.8950

Table 11. Multi-seed decomposition of the PatMamba gain (five seeds, paired t-test, df $= 4$). Δ is the relative change versus the baseline (rows 1 and 3) or the control (row 2); @all is the full validation set and @5+ the dense $\geq$ 5-panel bucket; * $p < 0.05$, n.s. not significant.

Comparison	Component	mIoU@all	mIoU@5+	IoU_1@5+	BB-IoU_1^{d3}@5+	hit@0.9@5+
PatMamba (grid 2×2) vs. baseline	architecture	-0.07% n.s.	$+0.99\%$ *	$+1.89\%$ *	$+1.52\%$ n.s.	-1.44% n.s.
PatMamba (grid 2×2, $w = 1.8$) vs. PatMamba (grid 2×2)	weighting	$+0.01\%$ n.s.	-0.10% n.s.	-0.16% n.s.	$+0.04\%$ n.s.	-1.61% n.s.
PatMamba (grid 2×2, $w = 1.8$) vs. baseline	total	-0.05% n.s.	$+0.89\%$ n.s.	$+1.73\%$ n.s.	$+1.56\%$ *	-3.03% *

Three findings follow. First, the grid-aware architecture is the significant contributor. Removing only the content-recall weighting, the control still improves the dense sub-figure IoU_1 over the baseline by $+1.89\%$ ($p = 0.023$) and the dense mIoU by $+0.99\%$ ($p = 0.027$), so the structure-aware gain is robust across seeds rather than an artifact of seed 42. Second, the content-recall weighting is in-distribution neutral. PatMamba differs from the control solely through the class weights, and across every metric in Table 11 the two groups are statistically indistinguishable; the weighting's value is, therefore, confined to the zero-shot cross-domain setting (Section 4.7), where it mitigates the content class under-segmentation under distribution shift, rather than to the in-distribution benchmark. Third, the gain carries a box precision cost. At the strict hit@$IoU_{0.9}$ threshold the total effect is -3.03% ($p = 0.018$), with comparable contributions from the architecture (-1.44%, n.s.) and the weighting (-1.61%, n.s.), consistent with a recall-oriented model that recovers content at the expense of high threshold box precision; the boundary-band metric BB-IoU_1^{d3} is nonetheless significantly improved ($+1.56\%$, $p = 0.033$), confirming that the boundary localization benefit survives the multi-seed test.

4.4. Ablation Studies

4.4.1. Grid Granularity

Table 12 ablates the grid size of the grid-aware scan. The 2×2 grid is optimal on the pixel metrics (0.918 mIoU), followed closely by 4×4 (0.916) and 3×3 (0.915); the two asymmetric partitions, 1×2 (0.914) and 2×1 (0.916), stay within 0.004 mIoU of the optimum, so the scan is also robust to the aspect ratio of the grid. The method is robust to the granularity and does not collapse as the grid becomes finer, while the coarsest 2×2 partition is also the cheapest; the detection hit rates differ by less than 0.5 points across the three symmetric granularities and by less than 0.7 points across all five partitions, i.e., within noise. We, therefore, adopt the 2×2 grid throughout.

Table 12. Ablation of the grid granularity.

Grid	mIoU	BF1	hit@IoU$_{0.7}$	hit@IoU$_{0.9}$
2×2	0.9178	0.7680	94.39%	90.85%
3×3	0.9148	0.7532	94.83%	90.53%
4×4	0.9163	0.7588	94.38%	90.72%
1×2 (asymmetric)	0.9139	0.7538	94.62%	91.16%
2×1 (asymmetric)	0.9156	0.7553	94.43%	90.80%

4.4.2. Gain Localization

To test whether the grid-aware scan acts through the layout structure as designed, Table 10 buckets the validation figures by their number of sub-figures and compares the grid 2×2 model against the baseline. The gain is concentrated almost entirely in the dense figures: single- and two-panel figures improve by less than $+0.2$ points of mIoU, three-panel figures regress (-0.1 mIoU and -0.4 IoU$_1$), and four-panel and five-or-more-panel figures improve by $+1.0$ and $+0.6$ points of mIoU and by $+1.3$ and $+1.7$ points of sub-figure IoU$_1$, respectively. The overall $+0.15$-point mIoU gain on the full validation set is, therefore, the expected dilution of a large, structure-specific effect by the 71.2% of single-panel figures, and not noise: the Boundary F1 improves consistently across all buckets. Figure 5 visualizes this localization.

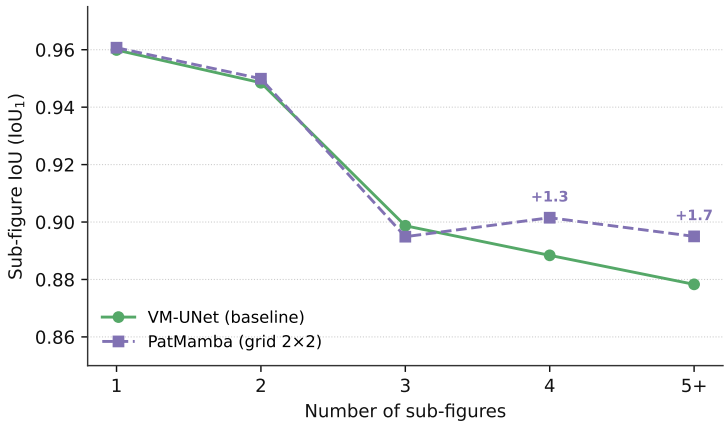


Figure 5. Localization of the grid-aware-scan gain: sub-figure IoU (IoU$_1$) as a function of the number of sub-figures, for the baseline and the grid 2×2 model (without content-recall weighting). The gain concentrates on the dense figures, with the $+1.3$-point gain on the four-panel bucket and the $+1.7$-point gain on the five-or-more-panel bucket, where the baseline's boundaries degrade most.

4.4.3. Direction Decomposition

The gain localization raises a natural question: does the dense-figure gain come from either grid direction alone, or from the two together? Table 13 decomposes the grid-aware

scan into its two appended directions. Appending either direction alone leaves the overall metrics essentially unchanged and does not reproduce the dense-figure gain: on four-panel figures the sub-figure IoU_1 moves by $+0.07$ and $+0.19$ points for π_4 and π_5 alone, and on five-or-more-panel figures by -0.30 and -0.32 points; all of these changes are within noise. By contrast, appending both directions yields $+1.31$ and $+1.67$ points. The two grid directions are, therefore, synergistic rather than additive: the layout prior is encoded only when both the row and the column traversal of the 2×2 partition are present, since a single grid axis degenerates to a line-wise reordering that the cross-scan already provides. Because each appended direction adds only the per-direction Δ projection (Section 3.4), this decomposition also rules out a capacity explanation, since the gain is attributable to the joint row/column grid structure rather than to merely adding scan directions. A six-direction control confirms this attribution: replacing the two grid directions with two diagonal, non-grid directions (at the same parameter count, Table 13, last row) yields a comparable, even marginally higher, aggregate mIoU (0.9188 versus 0.9178), so six directions of any kind saturate the aggregate score; however, this control does not reproduce the dense-figure localization, where the sub-figure IoU_1 on the five-or-more-panel bucket is 0.8675 against 0.8950 for the grid scan. The dense-figure gain is, therefore, specific to the grid layout prior rather than to the mere number of scan directions. Figure 6 visualizes this decomposition.

Table 13. Decomposition of the grid-aware scan into its two grid directions (validation set). IoU_1@4 and IoU_1@$\geq$5 denote the sub-figure IoU on the four-panel and five-or-more-panel buckets of Table 10.

Scan Orders	mIoU	BF1	IoU_1	IoU_1@4	IoU_1@$\geq$5
$\{\pi_0, \ldots, \pi_3\}$ (cross-scan)	0.9163	0.7606	0.9543	0.8884	0.8783
$+\pi_4$ (grid row-priority)	0.9155	0.7554	0.9543	0.8891	0.8753
$+\pi_5$ (grid column-priority)	0.9164	0.7649	0.9546	0.8903	0.8751
$+\pi_4 + \pi_5$ (grid 2×2)	0.9178	0.7680	0.9553	0.9015	0.8950
6-dir non-grid	0.9188	0.7687	0.9563	0.9063	0.8675

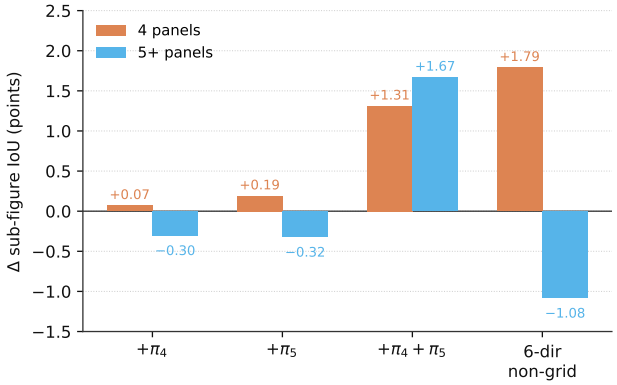


Figure 6. Direction decomposition of the grid-aware scan: the sub-figure IoU_1 gain over the four-way baseline on the four-panel and five-or-more-panel buckets. Appending either grid direction alone ($+\pi_4$ or $+\pi_5$) leaves the gain near zero, whereas appending both directions ($+\pi_4 + \pi_5$) yields $+1.31$ and $+1.67$ points, confirming that the two grid directions act synergistically. A 6-dir non-grid control (two diagonal, non-grid directions) improves the four-panel bucket ($+1.79$) but regresses the five-or-more-panel bucket (-1.08), showing that the dense-figure gain is specific to the grid layout prior rather than to the mere number of directions.

The augment design also governs the method's behavior when the fixed grid is misaligned with the actual arrangement, such as an irregular tiling, a rotated sheet, or a figure whose panels do not fall on a clean 2×2 lattice. Because the two grid directions are appended to the four whole image cross-scan directions rather than substituted for them

(Section 3.4), a misaligned partition never discards the global context: its grid directions merely become two additional, imperfectly aligned scan orders on top of the intact cross-scan. The six-direction non-grid control of Table 13 is precisely this limiting case (two extra directions with no layout alignment at all), and it leaves the aggregate mIoU marginally higher than the aligned grid (0.9188 versus 0.9178). Misalignment, therefore, degrades gracefully to the baseline; what is forfeited is only the dense-figure localization that the aligned grid contributes (sub-figure IoU_1 of 0.8675 versus 0.8950 on the five-or-more-panel bucket). The asymmetric partitions of Table 12 make the same point on the aspect ratio axis, remaining within 0.004 mIoU of the optimum. The 2×2 grid is thus not a hard structural assumption but a cheap prior that is well matched to the dominant layout, and the model remains correct, albeit somewhat less layout-aware, when that prior is wrong. A per figure adaptive partition that recovers the true panel tiling, in the spirit of dynamic scanning strategies [35], could recover the dense-figure gain on irregular layouts and is left to future work.

4.4.4. Content-Recall Weighting

The content-recall weight w of Section 3.6 is the only hyperparameter specific to PatMamba. Table 14 sweeps it over $\{1.6, 1.7, 1.8, 1.9\}$ and reports the in-domain sub-figure detection: both hit rates are essentially flat across the range (within 0.32 and 0.67 points, respectively), confirming that the weighting does not act through in-distribution accuracy and that the method is insensitive to the exact value. We fix $w = 1.8$, which attains the highest hit@$IoU_{0.7}$ at a hit@$IoU_{0.9}$ cost well within this noise; the benefit of the weighting is realized under distribution shift, as reported in Section 4.7.

Table 14. Sensitivity of the in-domain sub-figure detection to the content-recall weight w (validation set).

Content-Recall Weight w	hit@$IoU_{0.7}$	hit@$IoU_{0.9}$
1.6	94.61%	91.12%
1.7	94.40%	91.04%
1.8	94.72%	90.57%
1.9	94.70%	91.24%

4.5. Efficiency Analysis

A central motivation for adopting the SS2D backbone is its linear complexity, which matters at the high resolutions required for thin line-art. Because the mamba-ssm kernels are custom and the FLOP counts produced by generic profilers (thop/fvcore) are unreliable for them, we report the parameter count, the measured inference latency and the peak GPU memory as the efficiency indicators, following common practice in the Mamba literature. Table 15 reports these indicators; the measurements are obtained with the same 512×512 input on the same RTX 4090 GPU. The grid-aware scan increases the parameter count by only +0.56M (+2.0%) and the inference latency by +12.8 ms (+44%), the latter being the expected constant factor cost of six scan passes versus four; the peak memory is essentially unchanged. The +44% latency is offset by an accuracy contribution that is localized rather than uniform. On the held-out test set the overall gain is a modest +0.05 mIoU points, but this headline averages the 71.2% of single-panel figures, where the scan is by design neutral, with the dense multi-panel figures where its effect is concentrated (Table 10, validation set): the sub-figure IoU rises by +1.3 and +1.7 points on four- and five-or-more-panel figures and the Boundary F1 by +0.7 points, and the directional decomposition ablation (Table 13) shows this gain to arise from the joint grid structure rather than from noise. Because the two grid directions add essentially no gain on single-panel figures, a practical deployment could route single-panel figures through the four-direction baseline

and dense figures through the six-direction scan, thereby converting the fixed cost into an on-demand cost. This routing strategy is plausible but has not yet been evaluated end-to-end, and we leave its validation to future work. PatMamba, therefore, pays a constant factor latency for a localized, boundary-level gain concentrated on the hardest cases, rather than trading latency uniformly across the dataset.

Table 15. Efficiency comparison (parameters, inference latency and peak GPU memory, 512 × 512/batch 1/fp32).

Model	Params (M)	Latency (ms)	Peak Mem (GB)
U-Net	7.76	2.96	0.27
SegFormer (MiT-B0)	3.71	6.91	0.31
VM-UNet (baseline)	27.96	29.19	0.30
PatMamba (grid 2 × 2)	28.52	42.00	0.31

4.6. Qualitative Results

Figure 7 visualizes the effect of the grid-aware scan on representative dense compound figures. The three success rows show figures with four or more sub-figures in which the baseline merges adjacent panels, while the grid-aware scan recovers clean sub-figure boundaries (the sub-figure IoU_1, annotated on the prediction panels, rises from 0.12–0.47 to 0.91–0.98). The last row is a failure case in which the grid-aware scan still merges adjacent sub-figures (sub-figure IoU_1 of 0.26, down from 0.53 for the baseline), illustrating the residual difficulty of very dense, heavily annotated layouts. The four rows correspond, top to bottom, to the DeepPatent2 validation figures USD0609507-20100209-D00002, USD0615150-20100504-D00004, USD0623506-20100914-D00003 and USD0608623-20100126-D00001.

4.7. Cross-Domain Generalization

All results above are measured inside the DeepPatent2 distribution on which the models were trained. A segmentation model that only generalizes within design patent drawings is of limited practical value, because a real document understanding pipeline must also ingest utility patents, biomedical figures, chemical reaction schemes and engineering diagrams, drawn in very different visual grammars. We, therefore, close the experimental section with a zero-shot cross-domain study: the four checkpoints selected in Section 4.2.3 are applied as is, without fine-tuning, without domain adaptation and without per dataset binarization tuning, to four out-of-domain corpora, and their ability to localize the graphical content (class 1: sub-figures, molecule regions or circuit components) is measured with the content IoU and the box-level hit rate of Section 4.2.2. The text-label class (class 2) is not transferred here, because the external corpora annotate textual content under heterogeneous conventions that do not align with the patent reference numeral definition of Section 3.1; the study, therefore, isolates the class-1 content localization behavior shared by all four corpora. The purpose is not to maximize absolute accuracy but to characterize (i) how far the structure-aware, grid-aware inductive bias of PatMamba carries beyond design patents, and (ii) whether the pixel versus box divergence observed in Section 4.3.1 reappears under distribution shift.

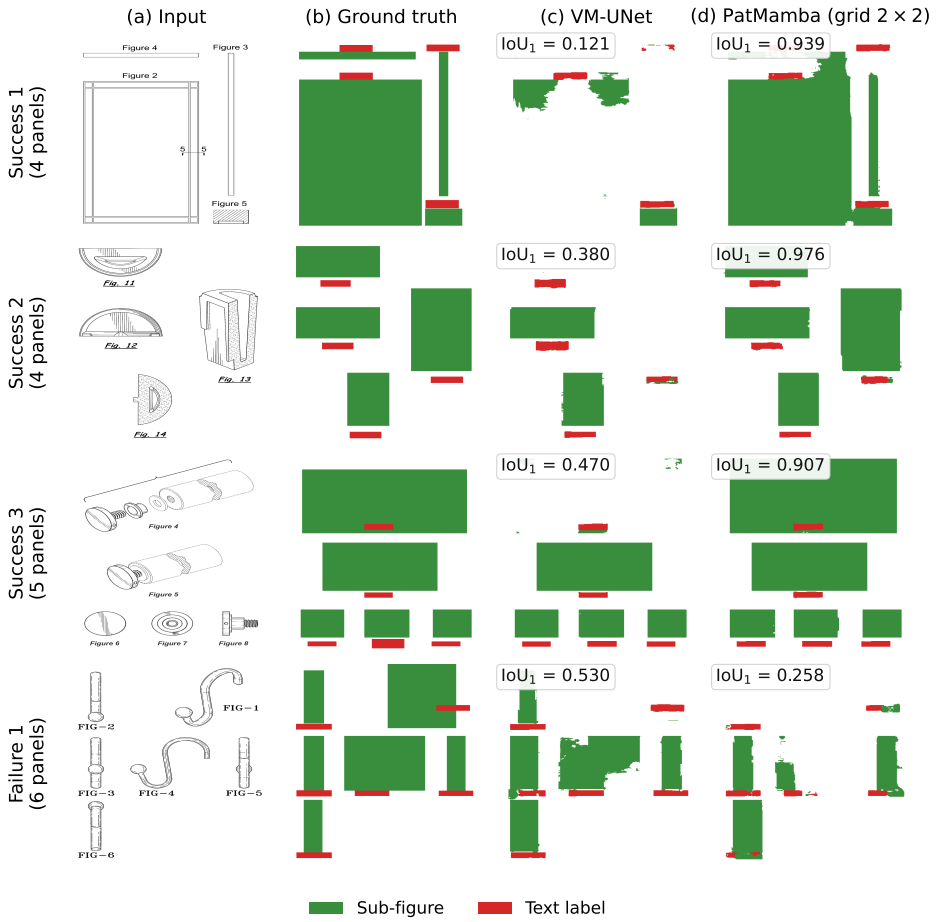


Figure 7. Qualitative comparison on dense compound figures: (**a**) input, (**b**) ground truth, (**c**) VM-UNet, (**d**) PatMamba with the grid 2 × 2 scan. The three success rows show dense figures (four or more sub-figures) where the grid-aware scan recovers clean sub-figure boundaries that the baseline merges (sub-figure IoU_1 annotated on the prediction panels); the last row is a failure case where the grid-aware scan still merges adjacent sub-figures.

The four target corpora are chosen to span a gradient of visual distance from the training domain, so that any decay can be attributed to domain rather than to an idiosyncrasy of a single dataset. The closest is E2-300 [44], a set of 300 US utility patent compound figures with sub-figure ground truth that we compiled and released for zero-shot evaluation; it shares the same line-art modality as the training design patents but is a different patent type. Biomedical compound figures (BioSci-Fig, introduced in FigEx [6]) retain the multi-panel layout but are produced by rasterizing photographed or screen-captured content. Chemical reaction schemes (RxnLabelData [45]) replace the panel layout with molecule structures surrounded by condition labels. The furthest corpus consists of handwritten circuit diagrams (CGHD [46]), which are dense, fine-grained and photorealistic in a completely different way. All four nonetheless share the core abstraction for which PatMamba was designed: localized graphical content entangled with textual annotations. Table 16 summarizes the four corpora and, in its last column, exposes the density gradient that is the dominant difficulty factor, which ranges from 2.4 sub-figures per patent figure to 6.0–6.4 per biomedical/chemical figure and 47.9 components per circuit diagram. For BioSci-Fig and RxnLabelData, the table reports the official release sizes; content localization is evaluated over the figures that carry content annotations (BioSci-Fig 43,183 boxes over 7174 evaluated figures; RxnLabelData 22,484 molecule boxes over 3492 of its 3500 figures, the remaining 8 containing text labels only).

Table 16. Cross-domain evaluation corpora used for zero-shot inference (no training or fine-tuning). "Content boxes" are the class-1 annotations: sub-figures for the patent and biomedical sets, molecule regions for the chemical set and component regions for the circuit set.

Dataset	Domain	Figures	Content Boxes	Boxes/Figure
E2-300 (this work)	Patent line-art (utility)	300	725	2.4
BioSci-Fig	Biomedical compound figures	7174	43,183	6.0
RxnLabelData	Chemical reaction schemes	3500	22,484	6.4
CGHD	Handwritten circuit diagrams	3291	157,608	47.9

Table 17 reports the content/sub-figure localization of the four frozen models (PatMamba with the content-recall weighting $w = 1.8$, the VM-UNet baseline, U-Net and SegFormer) on the four corpora at the shared 512×512 resolution. Every figure is converted to grayscale, resized to 512×512 and adaptively binarized through the same inference preprocessing, and the predicted class-1 region is post-processed into component boxes exactly as in Section 4.3.1; no label of any target corpus is seen at any stage. Three complementary behaviors emerge from the table.

Table 17. Zero-shot cross-domain content/sub-figure localization (class 1) at 512×512. IoU is the content region IoU; hit@$\text{IoU}_{0.7/0.9}$ are box-level hit rates following the protocol of Section 4.2.2.

Domain	Model	IoU	hit@$\text{IoU}_{0.7}$	hit@$\text{IoU}_{0.9}$
E2-300 (US utility patents)	U-Net	0.7811	37.24%	14.21%
	SegFormer (MiT-B0)	0.7476	70.34%	49.79%
	VM-UNet (baseline)	0.7859	73.93%	52.28%
	PatMamba (grid 2×2, $w = 1.8$)	0.7999	73.52%	55.31%
BioSci-Fig	U-Net	0.8539	1.65%	0.50%
	SegFormer (MiT-B0)	0.8040	10.35%	4.36%
	VM-UNet (baseline)	0.6427	9.69%	3.77%
	PatMamba (grid 2×2, $w = 1.8$)	0.7270	10.00%	3.93%
RxnLabelData	U-Net	0.4666	5.09%	0.90%
	SegFormer (MiT-B0)	0.3812	12.24%	4.37%
	VM-UNet (baseline)	0.3188	8.42%	2.36%
	PatMamba (grid 2×2, $w = 1.8$)	0.3831	10.11%	2.70%
CGHD	U-Net	0.0865	0.00%	0.00%
	SegFormer (MiT-B0)	0.0812	0.00%	0.00%
	VM-UNet (baseline)	0.0790	0.01%	0.00%
	PatMamba (grid 2×2, $w = 1.8$)	0.0846	0.02%	0.00%

Three conclusions follow. First, a domain distance gradient. PatMamba's content IoU decays monotonically as the corpus moves away from patent line-art, falling from 0.80 on utility patents and 0.73 on biomedical figures to 0.38 on reaction schemes and 0.08 on circuit diagrams. The model, therefore, transfers well to the visually closest domain and degrades in a regular, interpretable way rather than collapsing abruptly, indicating that the learned notion of "localized graphical content" is a genuine visual concept rather than a DeepPatent2-specific artifact. Second, pixel IoU and box localization are again decoupled. Exactly as in Section 4.3.1, U-Net attains the highest content IoU in all three non-patent domains (0.854, 0.467, 0.087) yet the lowest box hit rate in every domain (37.2%, 1.65%, 5.09% and 0.00% at $\text{IoU}_{0.7}$), because its blurred boundaries prevent clean component boxes; the pixel/box divergence is, therefore, a property of the CNN backbone's boundary fidelity and is not specific to the training data. Third, PatMamba's box-level advantage is confined to the closest domain with a panel structure. On utility patents it attains the highest content IoU (0.800) and the highest strict-threshold hit rate (55.3% at $\text{IoU}_{0.9}$, versus 52.3% for the baseline and 49.8% for SegFormer), while at the loose threshold it is tied with

the baseline (73.5% versus 73.9%); on the three non-patent corpora the box-level edge shifts to SegFormer, which attains the highest hit rates on the biomedical and chemical figures (10.35%/12.24% at $IoU_{0.7}$ versus 10.00%/10.11% for PatMamba). The structure-aware scan does not rescue the far domains, where the bottleneck shifts from panel separation to fine-grained instance density: the circuit corpus averages 47.9 components per figure, so its box-fill localization collapses to ≈ 0 for every model. Taken together, the study shows that PatMamba's structure-aware bias transfers to the visually closest domain where the panel-layout prior still holds, and that the remaining cross-domain gap is driven primarily by content density and granularity rather than by the panel-layout prior alone.

5. Discussion

5.1. Comparison with MedT

MedT reports a 99.5% figure hit rate on an instance-level, two-class task over about 4000 figures, whereas our task is a three-class pixel-level segmentation evaluated on a different, larger and denser subset. We report our models on the same figure-level box-IoU metric so that the two formulations can be compared on a common footing, and we additionally provide the pixel-level metrics that a purely instance-level method cannot produce. MedT is not reimplemented here because its two-class, instance-level formulation does not align with our three-class pixel-level task; its 99.5% figure is, therefore, quoted as reported by the original authors rather than reproduced. The relevant comparison within our protocol is across the backbone families (traditional, CNN, Transformer, Mamba), where the ordering is consistent and large.

5.2. Why mIoU Saturates

The small mIoU differences between the Mamba models reflect the composition of the metric rather than a ceiling on the method. The background class, which accounts for the large majority of pixels, is easy for the Mamba models ($IoU_0 \approx 0.94$), and its weight in the mean pins their mIoU differences within roughly ± 0.005. The discriminative information, therefore, lies in the figure-level detection metric and in the pixel metrics conditioned on class and density, where the differences are substantial (for example, the $+0.7$-point Boundary F1 of the grid-aware scan and its $+1.3/+1.7$-point sub-figure IoU gain on dense figures). The value of the grid-aware scan is not a headline mIoU improvement, but its localized improvement on box boundary alignment (Boundary F1 and BB-IoU, Section 4.2.2) and on the dense, hard cases.

5.3. Limitations and Future Work

Several limitations point to future work. First, the dataset is dominated by single-panel figures (71.2%), so the dense multi-panel figures on which the grid-aware scan matters most are comparatively scarce (273 and 137 figures in the buckets with four and with five or more sub-figures, respectively), and the localized gains should be consolidated on more data. Second, the ground truth rasterizes annotation bounding boxes into solid rectangles, so the pixel-level metrics measure box fitting accuracy rather than stroke-level fidelity; a stroke-level mask derived from the raw drawings would be needed to evaluate fine line preservation directly. Third, we evaluate at a 512×512 resolution; higher resolutions such as 768 are likely to help the finest boundaries and remain to be explored. Fourth, the text-label class remains the hardest ($IoU_2 \approx 0.86$), and coupling it with an OCR module could further improve both text separation and the figure-text association downstream. Fifth, the 2×2 grid is a fixed prior that can be misaligned with irregular tilings; as shown in Section 4.4, such misalignment degrades gracefully to the baseline rather than failing, and replacing the fixed grid with a per figure adaptive partition is a promising direction

for recovering the dense-figure gain on irregular layouts. Sixth, a runtime routing rule that disables the two grid directions for single-panel figures and enables them only for dense multi-panel figures would recover the baseline latency at no accuracy cost, as suggested by the per bucket results of Table 10; this rule has not yet been evaluated end-to-end and is left to future work. Finally, the end-to-end value of our decomposition for sub-figure retrieval and figure-text association has not yet been demonstrated and is a natural next step.

6. Conclusions

We presented PatMamba, a Vision Mamba framework for the pixel-level three-class segmentation of compound patent figures, and a grid-aware scan that encodes the compound-figure grid structure into the selective-scan directions. The grid-aware scan adds only $+0.56$M parameters ($+2.0\%$) but increases inference latency by $+44\%$ (42.00 ms versus 29.19 ms), the expected constant factor cost of six scan passes versus four. On DeepPatent2, the Mamba backbone lifts figure-level sub-figure detection from 70.7% for the traditional run-length-smoothing baseline to 94.5% hit@$\text{IoU}_{0.7}$; the grid-aware scan lifts the pixel mIoU to 0.918 and further improves the sub-figure IoU by $+1.3/+1.7$ points on four- and five-or-more-panel figures with a consistent $+0.7$-point Boundary F1 gain, which a directional decomposition ablation shows to arise from the joint action of the two grid directions rather than from either alone. Because 71.2% of the figures are single-panel figures on which the grid prior is neutral by design, the aggregate pixel-level gain is necessarily small ($+0.05$ points of test mIoU); the trade-off between accuracy and efficiency is, therefore, asymmetric, with the constant factor latency paying for a localized, boundary-level gain concentrated on the dense figures that are hardest to segment. In zero-shot transfer to utility patents, biomedical figures, chemical reaction schemes and handwritten circuit diagrams, the structure-aware bias persists on the visually closest domain and decays in a regular gradient with domain distance. By jointly segmenting the sub-figure and text-label regions, PatMamba provides a fine-grained foundation for patent figure understanding and the downstream tasks that depend on it.

Author Contributions: Conceptualization, Z.M. and F.Z.; methodology, Z.M.; software, Z.M.; validation, Z.M.; formal analysis, Z.M.; investigation, Z.M. and F.Z.; data curation, Z.M.; writing—original draft preparation, Z.M.; writing—review and editing, F.Z.; supervision, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The DeepPatent2 dataset is publicly available at https://www.nature.com/articles/s41597-023-02653-7 (accessed on 1 October 2026). The E2-300 dataset introduced in this work is publicly available at https://doi.org/10.5281/zenodo.23936087. The external cross-domain evaluation corpora (BioSci-Fig, RxnLabelData and CGHD) are publicly available from their respective publications.

Acknowledgments: The authors are grateful to all individuals who have contributed to this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BB-IoU	Boundary-Band Intersection over Union
BF1	Boundary F1

CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Network
DETR	Detection Transformer
FLOPs	Floating-Point Operations
GPU	Graphics Processing Unit
IoU	Intersection over Union
mDice	mean Dice coefficient
MedT	Medical Transformer
mIoU	mean Intersection over Union
OCR	Optical Character Recognition
RLSA	Run-Length Smoothing Algorithm
SS2D	2D Selective Scan (four cross-scan directions of VMamba)
SSM	State Space Model
VM-UNet	Vision Mamba U-Net

References

1. Kucer, M.; Oyen, D.; Castorena, J.; Wu, J. DeepPatent: Large Scale Patent Drawing Recognition and Retrieval. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 557–566. https://doi.org/10.1109/WACV51458.2022.00063.
2. Ajayi, K.; Wei, X.; Gryder, M.; Shields, W.; Wu, J.; Jones, S.M.; Kucer, M.; Oyen, D. DeepPatent2: A Large-Scale Benchmarking Corpus for Technical Drawing Understanding. *Sci. Data* **2023**, *10*, 772. https://doi.org/10.1038/s41597-023-02653-7.
3. Kim, H.; Gim, G. Enhancing Patent Document Similarity Evaluation and Classification Precision through a Multimodal AI Approach. *Appl. Sci.* **2025**, *15*, 9254. https://doi.org/10.3390/app15179254.
4. Shukla, S.; Sharma, N.; Gupta, M.; Mishra, A. PatentLMM: Large Multimodal Model for Generating Descriptions for Patent Figures. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 20488–20496. https://doi.org/10.1609/aaai.v39i19.34257.
5. Awale, S.; Müller-Budack, E.; Ewerth, R. Patent Figure Classification Using Large Vision-Language Models. In *Advances in Information Retrieval (ECIR 2025)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15573, pp. 20–37. https://doi.org/10.1007/978-3-031-88711-6_2.
6. Song, J.; Das, A.; Cui, G.; Huang, Y. FigEx: Aligned Extraction of Scientific Figures and Captions. In *Findings of the Association for Computational Linguistics: EMNLP 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 16558–16571.
7. Hoque, M.R.U.; Wei, X.; Choudhury, M.H.; Ajayi, K.; Gryder, M.; Wu, J.; Oyen, D. Segmenting Technical Drawing Figures in US Patents. In Proceedings of the AAAI-22 Workshop on Scientific Document Understanding (SDU 2022), CEUR-WS, Virtual, 1 March 2022; Volume 3164, pp. 1–6.
8. Wong, K.Y.; Casey, R.G.; Wahl, F.M. Document Analysis System. *IBM J. Res. Dev.* **1982**, *26*, 647–656. https://doi.org/10.1147/rd.266.0647.
9. Henry, D.J. System and Method for Search. U.S. Patent Application Publication No. US 2009/0228777 A1, 10 September 2009.
10. Duan, L.; Yuan, B.; Li, J.; Ma, W.; Yang, Z. An Intelligent Processing Method for Graphic and Text Information in Early Patent Document Scans. Chinese Patent No. CN 103927533 A, 18 June 2014.
11. Wang, Y.; Qiu, Y.; Chen, Y. PatentOCR: A Benchmark for Fine-Grained Detection and OCR of Component Identifiers in Patent Drawings. *Sci. Data* **2026**, *13*, 1395. https://doi.org/10.1038/s41597-026-07829-5.
12. Zhong, X.; Tang, J.; Yepes, A.J. PubLayNet: Largest Dataset Ever for Document Layout Analysis. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1015–1022. https://doi.org/10.1109/ICDAR.2019.00166.
13. Valanarasu, J.M.J.; Oza, P.; Hacihaliloglu, I.; Patel, V.M. Medical Transformer: Gated Axial-Attention for Medical Image Segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI 2021)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12901, pp. 36–46. https://doi.org/10.1007/978-3-030-87193-2_4.
14. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision (ECCV 2020)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13.
15. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
16. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; Liu, W.; Xiao, B. Deep High-Resolution Representation Learning for Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 5693–5703. https://doi.org/10.1109/CVPR.2019.00584.

17. Elyan, E.; Jamieson, L.; Ali-Gombe, A. Deep Learning for Symbols Detection and Classification in Engineering Drawings. *Neural Netw.* **2020**, *129*, 91–102. https://doi.org/10.1016/j.neunet.2020.05.025.
18. Bickel, S.; Goetz, S.; Wartzack, S. Symbol Detection in Mechanical Engineering Sketches: Experimental Study on Principle Sketches with Synthetic Data Generation and Deep Learning. *Appl. Sci.* **2024**, *14*, 6106. https://doi.org/10.3390/app14146106.
19. Jamieson, L.; Moreno-García, C.F.; Elyan, E. A Review of Deep Learning Methods for Digitisation of Complex Documents and Engineering Diagrams. *Artif. Intell. Rev.* **2024**, *57*, 136. https://doi.org/10.1007/s10462-024-10779-2.
20. Sauvola, J.; Pietikäinen, M. Adaptive Document Image Binarization. *Pattern Recognit.* **2000**, *33*, 225–236. https://doi.org/10.1016/S0031-3203(99)00055-2.
21. Souibgui, M.A.; Kessentini, Y. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1180–1191. https://doi.org/10.1109/TPAMI.2020.3022406.
22. Alaei, A.; Bui, V.; Doermann, D.; Pal, U. Document Image Quality Assessment: A Survey. *ACM Comput. Surv.* **2024**, *56*, 29. https://doi.org/10.1145/3606692.
23. Tarbă, N.; Boiangiu, C.-A.; Voncilă, M.-L. State-of-the-Art Document Image Binarization Using a Decision Tree Ensemble Trained on Classic Local Binarization Algorithms and Image Statistics. *Appl. Sci.* **2025**, *15*, 8374. https://doi.org/10.3390/app15158374.
24. Li, X.-H.; Yin, F.; Liu, C.-L. DocSAM: Unified Document Image Segmentation via Query Decomposition and Heterogeneous Mixed Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA , 10–17 June 2025; pp. 15021–15032.
25. Gu, A.; Goel, K.; Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces. In Proceedings of the Tenth International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
26. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In Proceedings of the First Conference on Language Modeling (COLM), Philadelphia, PA, USA, 7–9 October 2024. https://doi.org/10.48550/arXiv.2312.00752.
27. Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In *Proceedings of the 41st International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research*; PMLR: Vienna, Austria, 2024; Volume 235, pp. 62429–62442.
28. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; Liu, Y. VMamba: Visual State Space Model. In Proceedings of the Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024; pp. 103031–103063. https://doi.org/10.52202/079017-3273.
29. Shi, Y.; Dong, M.; Xu, C. Multi-Scale VMamba: Hierarchy in Hierarchy Visual State Space Model. In Proceedings of the Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024; pp. 25687–25708. https://doi.org/10.52202/079017-0808.
30. Liu, J.; Yang, H.; Zhou, H.-Y.; Xi, Y.; Yu, L.; Li, C.; Liang, Y.; Shi, G.; Yu, Y.; Zhang, S.; Zheng, H.; Wang, S. Swin-UMamba: Mamba-Based UNet with ImageNet-Based Pretraining. In *Medical Image Computing and Computer Assisted Intervention (MICCAI 2024)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2024; pp. 615–625. https://doi.org/10.1007/978-3-031-72114-4_59.
31. Ma, C.; Wang, Z. Semi-Mamba-UNet: Pixel-Level Contrastive and Cross-Supervised Visual Mamba-Based UNet for Semi-Supervised Medical Image Segmentation. *Knowl.-Based Syst.* **2024**, *300*, 112203. https://doi.org/10.1016/j.knosys.2024.112203.
32. Zhang, H.; Zhu, Y.; Wang, D.; Zhang, L.; Chen, T.; Wang, Z.; Ye, Z. A Survey on Visual Mamba. *Appl. Sci.* **2024**, *14*, 5683. https://doi.org/10.3390/app14135683.
33. Ruan, J.; Li, J.; Xiang, S. VM-UNet: Vision Mamba UNet for Medical Image Segmentation. *ACM Trans. Multimed. Comput. Commun. Appl.* 2025, *online first*. https://doi.org/10.1145/3767748.
34. Zhu, Q.; Fang, Y.; Cai, Y.; Chen, C.; Fan, L. Rethinking Scanning Strategies with Vision Mamba in Semantic Segmentation of Remote Sensing Imagery: An Experimental Study. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 18223–18234. https://doi.org/10.1109/JSTARS.2024.3472296.
35. Zhang, Z.; Ma, Q.; Zhang, T.; Chen, J.; Zheng, H.; Gao, W. Switch-UMamba: Dynamic Scanning Vision Mamba UNet for Medical Image Segmentation. *Med. Image Anal.* **2026**, *107*, 103792. https://doi.org/10.1016/j.media.2025.103792.
36. Huo, Y.; Li, P.; Chen, Y.; Gao, W. CrackSnakeNet: Multi-Directional Snake Scanning with Mamba for Topology-Preserving Crack Segmentation. *Constr. Build. Mater.* **2026**, *543*, 148064. https://doi.org/10.1016/j.conbuildmat.2026.148064.
37. Zhang, Y.; He, X.; Zhan, C.; Li, J. Visual State Space Model for Image Deraining with Symmetrical Scanning. *Symmetry* **2024**, *16*, 871. https://doi.org/10.3390/sym16070871.
38. Zhao, S.; Chen, H.; Zhang, X.; Xiao, P.; Bai, L.; Ouyang, W. RS-Mamba for Large Remote Sensing Image Dense Prediction. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5633314. https://doi.org/10.1109/TGRS.2024.3425540.
39. Zhang, Q.; Geng, G.; Zhou, P.; Liu, Q.; Wang, Y.; Li, K. Link Aggregation for Skip Connection–Mamba: Remote Sensing Image Segmentation Network Based on Link Aggregation Mamba. *Remote Sens.* **2024**, *16*, 3622. https://doi.org/10.3390/rs16193622.
40. Wang, T.; Tian, D.; Zhao, H.; Liu, J.; Wang, W.; Li, C.; Liu, G. Hierarchical Multi-Scale Mamba with Tubular Structure-Aware Convolution for Retinal Vessel Segmentation. *Entropy* **2025**, *27*, 862. https://doi.org/10.3390/e27080862.

41. Kervadec, H.; Bouchtiba, J.; Desrosiers, C.; Granger, E.; Dolz, J.; Ben Ayed, I. Boundary Loss for Highly Unbalanced Segmentation. *Med. Image Anal.* **2021**, *67*, 101851. https://doi.org/10.1016/j.media.2020.101851.
42. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 8748–8763.
43. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Virtual, 6–14 December 2021; pp. 12077–12090.
44. Ma, Z. E2-300: 300 U.S. Utility Patent Compound Figures for Zero-Shot Evaluation. *Zenodo* **2026**. https://doi.org/10.5281/zenodo.23936087.
45. Tang, J.; Wu, Z.; Xu, L. RobustRDP: Advancing Reaction Diagram Parsing via Synthetic-to-Real Data Scaling and Robustness-Oriented Training. In Proceedings of the Computer Vision—ECCV 2026, Malmö, Sweden, 8–12 September 2026.
46. Thoma, F.; Bayer, J.; Li, Y.; Dengel, A. A Public Ground-Truth Dataset for Handwritten Circuit Diagram Images. In *Document Analysis and Recognition—ICDAR 2021 Workshops*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; pp. 20–27. https://doi.org/10.1007/978-3-030-86198-8_2.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.