

O'REILLY®

Transformers

The Definitive Guide

Applications Beyond NLP



Nicole Koenigstein

"Most transformer books stop at language. Nicole Koenigstein keeps going—through vision, audio, video, time series, and RL—and makes the shared architecture feel inevitable rather than incidental. The production insights on KV caching and deployment trade-offs are especially hard-won."

Mike Erlihson
Head of AI, DriveNets

"The focus on reasoning, alignment, and system-level optimization captures exactly where the next wave of AI capabilities is heading. This book stands out for making these topics both accessible and highly actionable."

Alma Thomas, MBA
Enterprise data architect and principal data engineer, Americo Financial Life and Annuity Insurance Company

Transformers: The Definitive Guide

The vast potential of AI technology remains untapped in areas like audio, video, and complex data analysis. In fact, many of today's professionals find it challenging to apply AI innovations across these diverse domains due to a lack of guidance and practical implementations.

This comprehensive guide, tailored especially for intermediate to advanced ML engineers, data scientists, and researchers, fills the gap. Author Nicole Koenigstein guides readers through the versatile applications of transformer models, not only deepening theoretical understanding but also emphasizing actionable strategies for real-world applications. The book equips you with a grand unified theory for transformers—foundational insights that keep you on the cutting edge regardless of state-of-the-art model evolutions.

You'll discover how to apply:

- Transformers in nontext domains like image, video, and music generation
- Reasoning models, coding agents, and multi-agent architectures
- Training-time and test-time optimization strategies
- Production deployment, runtime engineering, and hardware efficiency

Nicole Koenigstein is an AI researcher and practitioner in agentic systems, working across research, consulting, teaching, and direct system implementation to build reliable, production-ready AI systems. She is the author of several books, including *AI Agents: The Definitive Guide* (O'Reilly, 2026), and has served as an external evaluator for a European Commission AI Grand Challenge.

DATA

US \$79.99 CAN \$99.99

ISBN: 978-1-098-16701-1



Transformers: The Definitive Guide

Applications Beyond NLP

Nicole Koenigstein

O'REILLY®

Transformers: The Definitive Guide

by Nicole Koenigstein

Copyright © 2026 Nicole Koenigstein. All rights reserved.

Published by O'Reilly Media, Inc., 141 Stony Circle, Suite 195, Santa Rosa, CA 95401.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Acquisitions Editor: Nicole Butterfield

Development Editor: Sarah Grey

Production Editor: Elizabeth Faerm

Copyeditor: Piper Content Partners

Proofreader: Dave Awl

Indexer: BIM Creatives, LLC

Cover Designer: Susan Brown

Cover Illustrator: José Marzan Jr.

Interior Designer: David Futato

Interior Illustrator: Kate Dullea

April 2026: First Edition

Revision History for the First Edition

2026-03-25: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781098167011> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Transformers: The Definitive Guide*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

The views expressed in this work are those of the author and do not represent the publisher's views. While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-098-16701-1

[LSI]

Table of Contents

Preface.....	ix
1. From First Principles to State-of-the-Art Transformers.....	1
Transformer Basics	2
Tokenizer: Text Representation in the Transformer	2
Token and Positional Embeddings	3
Attention Mechanism	5
Encoder and Decoder Parts	7
Enhancements in Transformer Design:	
Longer Context and Attention Variations	9
Longer Context Windows with Better Performance	9
Attention Mechanism Variations	16
Conclusion	25
2. Transformers for Time Series.....	27
Understanding the Intricacies of Time Series Data	28
Autocorrelation and Partial Autocorrelation	28
Cointegration	29
Cross-Correlation	30
Stationarity	31
Trend and Seasonality	31
Preparing a Dataset	32
Time Series Modeling in Various Application Domains	32
Tokenizing Time Series Data	33
Chronos: Learning the Language of Time Series	38
Fine-Tuning Chronos	40
PatchTST: A Time Series Is Worth 64 Words	43
Fine-Tuning PatchTST on Historical IBM Stock Prices	45

TimesFM: A Decoder-Only Time Series Foundation Model	48
Fine-Tuning TimesFM on Hourly Energy Consumption Data	53
AnomalyBERT for Self-Supervised Anomaly Detection	55
Conclusion	58
3. Transformers for Vision Tasks.....	59
Overview of Different Vision Tasks	60
Embeddings and Tokenization for Vision Models	61
Key Strategies for Improving the Robustness and Effectiveness of Vision Tasks	64
Swin Transformer V2	66
Image classification with Swin Transformer V2	68
Segment Anything	72
Fine-Tuning SAM on a Custom Dataset	76
Segment Anything in Images and Videos	83
Segment Videos and Images with Concept Prompts	88
Conclusion	95
4. Transformers for Image Generation.....	97
Introduction to Generative Image Models	98
Diffusion Models: What's That Noise About?	99
Classifier-Free Guidance in Diffusion Models	101
Scalable Diffusion Models with Transformers	102
Generating Images with the DiT	106
PIXART- α	108
PixArt- Σ	111
Generating Images with PixArt- Σ	115
Diffusion Vision Transformers for Image Generation	120
Interpretable Features with Diffusion Transformers	123
Conclusion	127
5. Transformers for Video Generation.....	129
Hidden Effectiveness of Latent Diffusion	130
LTX-Video: Video in Realtime	132
Latte: Structured Detail, Poured into Every Video Frame	137
Tora: From Trajectory to Storyline, One Frame at a Time	140
Conclusion	148
6. From Sound to Token and Back: Transformers in the Audio Domain.....	149
From Waveforms to Spectrograms:	
Understanding the Structure of Audio Data	151
Audio as a Waveform	151

Sampling Rate and the Nyquist Theorem	152
Amplitude, Bit Depth, and Quantization	152
The Frequency Domain and Fourier Transform	153
Spectrograms and the Short-Time Fourier Transform	154
The Mel Spectrogram and Perceptual Scaling	155
Phase, Reconstruction, and Vocoders	155
Audio Modeling in Various Application Domains	159
Transformer Architectures for Audio:	
From Perception to Foundational Intelligence	160
The Rise of Speech Transformers: The Impact of Whisper	160
Audio Foundation Models: Unifying Understanding, Generation, and Conversation	163
Qwen2-Audio	164
Transcribing a Meeting with Kimi-Audio	168
Segment Anything in Audio	174
Beyond Text and Speech: Transformers as Music Composers	177
Conclusion	181
7. Reinforcement Learning Transformers.....	183
Getting Started with Reinforcement Learning	183
Foundational Concepts in Reinforcement Learning	185
Online and Offline Reinforcement Learning	185
Model-Based and Model-Free Approaches	186
On-Policy Versus Off-Policy Reinforcement Learning	186
Temporal Difference Learning	188
World Models in Reinforcement Learning	188
Transformers in Reinforcement Learning	189
Decision Transformer	190
Going Live: Online Decision Transformer	196
A Brave New World: Stochastic Transformer-Based World Model	199
TWISTER: Transformer-Based World Models with Contrastive Predictive Coding	204
Conclusion	211
8. Embracing the Era of Experience: Transformers for Planning, Reasoning, and Coding	213
From Human Data to Lived Experience	214
Learning to Reason: From Pretraining to Reinforcement Learning	216
DeepSeek-R1: Reinforcing Reasoning Capabilities	217
Qwen3: Unified Reasoning with Dynamic Control	220
Qwen3-Coder: Agentic Reasoning for Open-Ended Coding	225
Kimi K2: Open Agentic Intelligence at Scale	227
Muon: Scaling Optimization for the Agentic Era	228

Inference with Kimi K2	229
Scaling Reasoning at Test-Time: Smarter, Not Just Bigger	230
Adaptive Branching Monte Carlo Tree Search (AB-MCTS)	232
The RethinkMCTS Framework for Code Generation	238
The S* Framework for Code Generation	239
Conclusion	241
9. From Scripts to Thinking: AI Agents for Complex Tasks.	243
Autonomy: What's Possible at the Moment?	244
Designing Agent Workflows	246
Multi-Agent Architectures	249
Agentic Communication: The Right Context Is All You Need	255
Beyond Context: How to Help Agents Remember	258
Agent Memory Types	258
Going Global and Lifelong	259
The Human Factor: Steering Agent Actions	261
Common Patterns for Human-in-the-Loop	261
Solving GitHub Issues with Coding Agents	268
Conclusion	271
10. Smarter, Better, Faster, Stronger: Optimizing LLMs and AI Agents.	273
Training-Time Intelligence: Reinforcement Learning for Agents	274
Beyond Hand-Crafted Rewards: How RULER Works	275
Training in Practice: ART in a Market Scenario	276
Reason Smarter, Not Harder: Adaptive Compute Allocation	285
The Delta Incentive: Enforcing Efficiency	285
Open Innovation: Community-Driven RL Frameworks	287
The Checkpoint Engine: Systems-Level Optimization for LLM Policy Updates	297
Conclusion	299
11. Deploying Transformer Models.	301
Choosing Between Open and Closed Source	302
Understanding the Architecture You're Deploying	303
Deploying Decoder-Only Models	305
Runtime Engineering for Decoder-Only Models	306
Security Considerations for Decoder-Only Deployments	307
Building Applications with Coding Models	308
Evaluating LLM Deployments in Production	313
Cost Efficiency and Hardware Comparison	314
Quantization	320
Test-Time Low-Rank Adaptation in Vision-Language Models	326
Conclusion	328

12. Where to Go Next: From Models to Intelligent Systems. 331
 Combining Capabilities: SAM 3 Agent 333
 The Science of Scaling Agentic Systems 339
 Conclusion 341

Index. 343

Preface

When I first started working with transformers in 2019, I was immediately drawn to them. Two years later, I built my own deep learning architecture based on attention, and that same year I taught my first university lecture on vision transformers. What struck me then, and continues to do so today, is just how versatile this architecture really is. Once you understand how transformers work, patterns begin to emerge. You start to see how the same principles apply across images, video, audio, and beyond. The deeper insight is not found in treating each domain as a separate problem, as is often the case in deep learning, but in recognizing how naturally the transformer architecture extends across domains.

As it turns out, it doesn't matter whether you're modeling discrete tokens, image patches, or spectrogram frames. The core logic remains the same. You may change the axis of representation, but the underlying architecture stays intact. That's the elegance of transformers, at least to me, and the reason I decided to write this book: to share my passion for transformers and let you discover that elegance at your own pace.

What also continues to fascinate me is what happens when transformers are used as part of a larger system. Their real power shows up when you combine multiple models into an agent that can reason, act, and iterate on a task. Beyond simply connecting models, my recent focus has been on pairing transformers with test-time compute and reinforcement learning. Together, these ideas make it possible to build systems that adapt, improve, and learn from their own behavior. That's what led me to focus my PhD research on self-improving AI agents.

As you read through this book, you'll follow both the development of transformers across different domains and my own path working with them, up to my most recent work on building AI agents. Along the way, I'll share how I personally think about these models—the mental models I use to reason about architectures, trade-offs, and system behavior and how those models shape the design decisions I make in practice. My goal isn't just to show you what works but to help you see why it works.

By the end of the book, my hope is that you arrive at the same place I did: seeing transformers not as isolated models but as building blocks inside larger systems. Transformers aren't just another architecture. They're a framework for abstraction. That's why they aren't a passing trend but a shift in how we generalize and connect modeling across domains. Arthur C. Clarke once said:

Any sufficiently advanced technology is indistinguishable from magic.

I hope, as you read through the book, that it gives you the tools to turn the “magic” of attention into a reliable, engineered reality.

What This Book Is About

This book is a practical systems guide for building intelligent transformer-powered applications across modalities, written by someone who has designed, debugged, and deployed these architectures in real products.

You'll start with first principles: tokenization, embeddings, attention, encoder and decoder design, and modern architectural enhancements for long-context reasoning. From there, the book expands across domains, showing how the same core transformer abstraction applies to time series forecasting, computer vision, image and video generation, audio understanding, reinforcement learning, and reasoning-driven coding systems.

Rather than treating each modality in isolation, the chapters build on shared structure: how tokens become patches, frames, or spectrogram slices; how attention adapts across domains; and how architectural choices propagate into performance, stability, and cost.

Later chapters move into reinforcement learning, test-time compute, reasoning models, and AI agents, connecting transformers with world models, planning algorithms, and multi-agent architectures. The goal is to help you understand how transformers evolve from standalone predictors into components of adaptive systems.

Beyond individual models, this book focuses on systems-level thinking. You'll learn how transformers behave when deployed in production, how to optimize inference and training pipelines, how to manage memory and compute, and how to integrate models into agentic workflows that can plan, reflect, and improve over time.

Throughout the book, theory is paired with concrete implementations, production considerations, and design trade-offs drawn directly from real deployments.

What This Book Is Not

This is not an introductory deep learning book or an introduction to large language models. I assume you already understand neural networks, backpropagation, basic machine learning workflows, and large language models. While core concepts are explained, the focus is on how transformers operate across domains and in practice.

This is also not a prompt engineering guide or a collection of model or API recipes. Although modern foundation models appear throughout the book, the emphasis is on understanding fundamentals. Model names are ephemeral. Instead, this book teaches the abstraction behind multimodal systems. My goal is to give you a kind of grand unified theory for transformers that remains relevant regardless of which model happens to be state of the art. Along the way, it focuses on training dynamics, optimization, and deployment, not on crafting prompts or consuming hosted endpoints.

This book is also not an academic exercise. While I include math where it's needed, I ensure not to overindulge. You'll see equations and architectural details throughout the book; they're always paired with practical engineering explanations. If you're looking for exhaustive theoretical proofs or benchmark-only discussions, this book takes a different, more applied approach.

Who This Book Is For

This book is tailored for intermediate to advanced machine learning (ML) engineers, data scientists, and AI architects who are ready to look beyond language models. It's for those who find themselves at the intersection of research and engineering: people who need to understand the why of an architecture to solve the how of a production problem.

Specifically, I wrote this for:

- The architect who needs to design systems that handle data beyond just text, such as video and image generation, image and audio classification, or time-series forecasting
- The engineer tasked with moving a model from a local notebook into a high-performance production environment and handling the production “gotchas,” such as spiking memory usage while optimizing key-value (KV) cache in decoder-only models
- The systems thinker interested in how individual transformers can be coordinated into AI agents that learn from their own test-time experience

Throughout these chapters, I don't shy away from the complexity and math that a technical book demands. However, I am a firm believer that technical complexity should always serve a practical purpose and be accompanied with clear coding examples. I share production insights: hard-won lessons I've gathered from building attention-based architectures and deploying them in the wild, so you don't have to. These are the nuances that research papers often leave out but that make the difference between a model that works in a demo and one that survives in production.

Navigating This Book

The book is organized to move from foundations to increasingly complex systems.

Chapter 1, “From First Principles to State-of-the-Art Transformers”, establishes transformer fundamentals, including tokenization, embeddings, attention, encoder and decoder architectures, and modern enhancements for long-context modeling.

Chapter 2, “Transformers for Time Series”, applies transformers to time series, covering domain-specific challenges such as stationarity and autocorrelation, and introduces foundation models for forecasting and anomaly detection.

Chapter 3, “Transformers for Vision Tasks”, focuses on vision tasks, including classification and segmentation, with practical examples using modern vision transformers.

Chapter 4, “Transformers for Image Generation”, and **Chapter 5, “Transformers for Video Generation”**, move into generative modeling for images and video, exploring diffusion transformers and latent representations for scalable generation.

Chapter 6, “From Sound to Token and Back: Transformers in the Audio Domain”, covers audio, from waveform fundamentals to speech, multimodal audio models, and music generation.

Chapter 7, “Reinforcement Learning Transformers”, introduces reinforcement learning with transformers, including decision transformers and world models.

Chapter 8, “Embracing the Era of Experience: Transformers for Planning, Reasoning, and Coding”, shifts toward reasoning and coding, examining how transformers learn to plan, reason, and solve open-ended problems using reinforcement learning and test-time compute.

Chapter 9, “From Scripts to Thinking: AI Agents for Complex Tasks”, builds on this by introducing AI agents, multi-agent architectures, memory systems, and human-in-the-loop workflows.

Chapter 10, “Smarter, Better, Faster, Stronger: Optimizing LLMs and AI Agents”, focuses on optimization, including reinforcement learning for agents, adaptive compute, and systems-level training strategies.

Chapter 11, “Deploying Transformer Models”, addresses deployment: runtime engineering, quantization, evaluation, security, and cost-aware production design.

Chapter 12, “Where to Go Next: From Models to Intelligent Systems”, looks forward, integrating models into intelligent systems and discussing how agentic architectures scale.

Each chapter is largely self-contained, so you can read selectively depending on your interests. However, the material builds conceptually, and reading sequentially provides the strongest mental model.

Now it’s your turn. From first principles and multimodal foundations to reinforcement learning, agentic systems, optimization, and deployment, the system is yours to design. Transformers are tools, and the chapters ahead are built around one goal: reliability beyond the demo stage.

The difference between a model that impresses in a notebook and one that survives in production lies in the details, in architectural choices, memory constraints, training dynamics, evaluation strategy, and system design. This book gives you those details. What you build next is where your own journey with transformers begins.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.



This element signifies a tip or suggestion.



This element signifies a general note.



This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at <https://oreil.ly/github-transformers>.

If you have a technical question or a problem using the code examples, please send email to support@oreilly.com.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Transformers: The Definitive Guide* by Nicole Koenigstein (O'Reilly). Copyright 2026 Nicole Koenigstein, 978-1-098-16701-1.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Online Learning



For more than 40 years, O'Reilly Media has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit <https://oreilly.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
141 Stony Circle, Suite 195
Santa Rosa, CA 95401
800-889-8969 (in the United States or Canada)
707-827-7019 (international or local)
707-829-0104 (fax)
support@oreilly.com
<https://oreilly.com/about/contact.html>

We have a web page for this book, where we list errata and any additional information. You can access this page at <https://oreil.ly/transformers-the-definitive-guide>.

For news and information about our books and courses, visit <https://oreilly.com>.

Find us on LinkedIn: <https://linkedin.com/company/oreilly>.

Watch us on YouTube: <https://youtube.com/oreillymedia>.

Acknowledgments

Writing a book or pursuing a career in a field such as AI is never truly a solitary effort. This feels especially true today, as the pace of progress in AI continues to accelerate, and the exchange of ideas becomes increasingly important, even though much of the work itself still happens during long, solitary days of research, coding, and writing. I would like to take this moment to thank the people who supported and inspired me at important moments throughout this journey.

In this spirit, I would like to thank colleagues, peers, students, readers, and course participants who engage with my work. Through discussions, questions, collaborations, and a shared curiosity about AI, these exchanges provide perspective and inspiration that shape how ideas develop and mature over time.

Special thanks go to Nicole Butterfield for her continued support during some of the more challenging moments while finishing this book. I would also like to express my appreciation to the rest of the team at O'Reilly who helped guide this book from manuscript to publication. In particular, I am grateful to Sarah Grey and Liz Faerm for their thoughtful support throughout the development process and for making my work as an author both easier and genuinely enjoyable. I would also like to thank the many members of the production team whose work behind the scenes helps bring a book like this to life.

Finally, I would like to thank the technical reviewers, Chris Fregly, Al Krinker, and Dibyendu Roy Chowdhury. Your careful review, questions, and feedback helped clarify important points and strengthen the overall quality of the book.

From First Principles to State-of-the-Art Transformers

Since its introduction in 2017, the transformer architecture¹ has revolutionized the field of natural language processing (NLP), marking a paradigm shift toward models capable of natural language understanding (NLU). This shift was possible because transformers process sequential data in parallel, enabling a deeper and more contextual understanding of language than was achievable with previous sequential models, like long short-term memory (LSTM) networks.

In recent years, transformers have evolved to impact a wide array of domains, including computer vision, speech recognition, reinforcement learning, and mathematical operations, moving beyond their initial usage within NLP. Their adaptability has led to significant advancements in machine translation, allowing for context-aware translations, and in scientific research, notably in predicting protein structures with remarkable accuracy.

Among the most exciting developments are reasoning models, which are advanced large language models (LLMs) trained with reinforcement learning to perform complex, multi-step reasoning. They generate internal chains of thought before answering, which is inspired by the human thought process. This technique first solves intermediate steps before getting to the final answer.

I assume in this book that you have at least some familiarity with the transformer architecture. Perhaps you've read the book *Natural Language Processing with Transformers* (O'Reilly), or a similar work. Moreover, I take it you're not just curious about

¹ Ashish Vaswani et al. "Attention Is All You Need" (2017).

transformers. You're here because you want to build real applications with transformers, and you want to do it right.

This chapter provides a focused review of the transformer architecture to set the stage for the more advanced and complex models beyond NLP that I'll cover in later chapters.

I'll begin with the basic transformer architecture, then explain how longer context becomes possible, and finish with a tour of various attention mechanisms. Throughout this chapter, and the ones that follow, I'll share practical insights from real deployments so that you can benefit from my experience and learn the patterns, pitfalls, and principles that matter when theory meets the hard surface of production.

Transformer Basics

This section explains the main architectural components of the original transformer model, such as encoder and decoder, positional embeddings, and attention mechanism.

The transformer architecture was originally developed for machine translation, a challenging sequence-to-sequence task in which the concept of tokenization plays a critical role. *Tokenization* breaks down sequences like sentences into manageable units, or tokens, that the transformer can effectively process. For example, in the sentence:

The Transformer has revolutionized NLP.

The word *the* represents a single word-level token.

Before we dive into the architectural components, understanding tokenization is crucial, as it facilitates the transformer's ability to interpret text. And it sets the foundation for its application to other sequences.

Tokenizer: Text Representation in the Transformer

A *tokenizer* is used to tokenize the text. This is the first step to make natural language digestible for the model, before applying token embeddings and finally positional embeddings. The different types of tokenization are:

Character-level tokenization

Character-level tokenization splits the underlying alphabet into each existing character in the sequence. If you used character-level tokenization for:

"The Transformer has revolutionized NLP."

it would yield:

[T, h, e, , ' , ... 'N, L, P, .]