The LLM Ladder

A Plain-English Guide from Tokens to Logits

A compact print companion for the canonical web guide. The web page remains the primary experience; this PDF is optimized for offline lookup, review, and table-of-terms scanning.

VERSION	AUTHOR	CANONICAL
v0.1 Working Reference	Grayson Dodson	graysond.xyz/research/llm-ladder/

CORE THESIS

An LLM is a learned mathematical system that produces probability distributions over possible next tokens given context.

TEXT -> TOKENS -> TOKEN IDS -> EMBEDDINGS -> TRANSFORMER LAYERS -> ATTENTION -> HIDDEN STATES -> LOGITS ->
SOFTMAX -> PROBABILITY DISTRIBUTION -> SAMPLING OR GREEDY DECODING -> NEXT TOKEN -> REPEAT

KEY MISCONCEPTION CALLOUTS

Temperature changes randomness, not truth.
Temperature reshapes the probability distribution before sampling. Lower temperature can make output more stable, but it does not guarantee factual accuracy.

Attention is not consciousness.
Attention is matrix math that assigns relationship weights between token representations. It is not awareness, intent, or human focus.

Context is not memory.
Context is what the model can currently see. Memory is stored information that may be brought back into context later.

Weights are not a document database.
Weights are learned parameters distributed across the model. They are not a searchable archive of source documents.

Softmax converts scores into probabilities.
Logits are raw scores. Softmax turns those scores into positive probabilities that sum to 1.

The model generates one token at a time, then repeats.
Each generated token is appended to context, which changes the next probability distribution.

Printable Term Index

LEVEL 0

AI in Plain English

Basic public-friendly concepts. This level gives the basic picture without internal mechanics.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
AI	Artificial intelligence is software that uses data, patterns, and computation to perform tasks that normally require judgment, perception, prediction, classification, generation, or decision support.	**Precision:** A large language model is one kind of AI, not all of AI.
Model	A model is a system that has learned patterns from data and can use those patterns to make predictions or generate outputs.	**Example:** Weather model -> predicts weather. Fraud model -> predicts suspicious transactions. Language model -> predicts possible next tokens.
Data	Data is information stored in a form that can be processed: text, numbers, images, audio, video, logs, documents, transactions, or code.	-
Training	Training is the process of adjusting a model using data. The model predicts, measures error, and updates internal parameters to become less wrong over time.	**Flow:** make prediction -> measure error -> adjust internal values -> repeat
Inference	Inference is using a trained model to produce an output. When a user prompts ChatGPT, that is inference.	**Precision:** During ordinary inference, the model is not learning new weights. It is using existing learned weights plus current context.
Prediction	A prediction is the model's estimate of what output is most likely or appropriate. For an LLM, the immediate prediction is usually what token should come next.	-
Probability	Probability is a number representing how likely something is, usually ranging from 0 to 1 or 0% to 100%.	**Example:** 0 = impossible. 1 = certain.
Hallucination	A hallucination is when a model produces information that sounds plausible but is false, unsupported, or fabricated.	**Precision:** Lowering temperature may reduce randomness, but it does not guarantee truth.
Grounding	Grounding ties a model's output to reliable evidence such as documents, databases, tools, search results, verified records, user files, or observed facts.	-

LEVEL 1

LLMs as Next-Token Systems

How LLMs generate text. This level explains the first major jump: LLMs generate text by predicting tokens.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
LLM	LLM means large language model. A language model is trained to predict or generate language.	-
Token	A token is a unit of text represented internally by an integer ID. It may be a word, part of a word, punctuation, whitespace, or a code fragment.	**Example:** unbelievable may be split into multiple tokens depending on the tokenizer.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Tokenizer	A tokenizer converts raw text into tokens and token IDs.	**Precision:** Different models can use different tokenizers, so the same text may split differently across models. **Example:** The dog ran. -> tokens -> token IDs
Token ID	A token ID is the integer assigned to a token. The number itself does not carry human meaning; it is an index used to look up a learned vector.	**Example:** Hello -> 15496
Vocabulary	The vocabulary is the full set of tokens a model can recognize and generate. A 100,000-token vocabulary means roughly 100,000 raw scores at each generation step.	-
Context	Context is the information currently available to the model: instructions, messages, conversation history, retrieved documents, tool outputs, and previously generated tokens.	-
Context Window	The context window is the maximum number of tokens the model can use at one time.	**Precision:** Tokens are not the same as words. In English, 128K tokens may roughly correspond to around 90K-100K words depending on tokenizer and text type.
Prompt	A prompt is the input given to a model. Prompting changes context, and changing context changes future probabilities.	-
Completion	A completion is the generated continuation or answer. In chat systems, the completion is usually the assistant response.	**Flow:** prompt -> completion

LEVEL 2

Representations, Vectors, and Embeddings

How text becomes math. This level explains how text becomes math the model can operate on.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Representation	A representation is how information is encoded inside the model. Humans see words; the model works with mathematical representations.	-
Scalar	A scalar is one number.	**Example:** 7, 0.5, -12.4
Vector	A vector is an ordered list of numbers representing a point, direction, or state in a mathematical space.	**Example:** [1.2, -0.8, 0.4]
Dimension	A dimension is one component or axis of a vector. LLM vectors can have hundreds or thousands of dimensions.	**Example:** [1.2, -0.8, 0.4] has three dimensions.
Matrix	A matrix is a two-dimensional array of numbers. Neural networks rely heavily on matrix multiplication.	**Precision:** A specific matrix has defined dimensions. It does not need to be square.
Tensor	A tensor is the general term for a multi-dimensional array.	**Example:** scalar = one number; vector = one-dimensional array; matrix = two-dimensional array; tensor = general multi-dimensional array.
Embedding	An embedding is a learned vector representation of a token, text chunk, image, or other input.	**Flow:** token ID -> embedding lookup -> vector
Embedding Table	An embedding table is a learned lookup table that maps token IDs to embedding vectors.	**Example:** token ID 15496 -> vector for that token

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Semantic Meaning	Semantic meaning refers to meaning or conceptual content. In embedding spaces, related terms may be represented near one another.	-
Latent Space	Latent space is a hidden mathematical representation space where learned relationships exist. Latent means hidden or not directly observed.	-

LEVEL 3

Transformers and Attention

How transformers process context. This level explains how the model processes token representations.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Architecture	Architecture means the design or structure of a model. For modern LLMs, the dominant architecture is the transformer.	-
Transformer	A transformer is a neural network architecture built around attention. Transformers are the foundation of most modern LLMs.	-
Layer	A layer is one processing stage inside a neural network. Large models stack many layers.	-
Transformer Block	A transformer block is one repeated unit inside a transformer model, typically including attention, an MLP, normalization, and residual connections.	-
Attention	Attention computes weighted relationships between token representations so the model can combine relevant context when forming internal representations.	**Precision:** Attention is not a conscious spotlight. It is matrix math producing weights between token representations.
Query, Key, and Value	Attention is often described using query, key, and value vectors.	**Example:** Query = what this token position is looking for. Key = what each other position offers. Value = information carried by that position.
Attention Score	An attention score is a raw compatibility score between a query and a key.	-
Attention Weight	An attention weight is a normalized importance value that determines how much information from a token position should be pulled into the current representation.	-
Attention Head	An attention head is one attention mechanism inside a transformer layer. Different heads can learn different relationship patterns.	-
Multi-Head Attention	Multi-head attention means the model uses multiple attention heads in parallel, letting it process multiple relationship patterns at the same time.	-
Feed-Forward Network / MLP	The feed-forward network, often called an MLP, transforms information within each token representation.	**Example:** Attention mixes information across token positions. The MLP transforms information within each token representation.
Activation Function	An activation function adds nonlinearity to a neural network. Examples include ReLU, GELU, and SiLU.	-

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Nonlinearity	Nonlinearity means the model can learn complex relationships instead of only straight-line relationships.	-
Residual Connection	A residual connection lets information skip around part of a layer, preserving useful information and making deep networks easier to train.	-
Layer Normalization	Layer normalization helps keep hidden states stable as they pass through many layers.	-
Positional Information	Positional information tells the model where tokens are in the sequence.	**Precision:** Without positional information, the model would struggle to distinguish token order.
RoPE	RoPE means rotary positional embedding. It is a common method for encoding token position in modern LLMs.	-
Hidden State	A hidden state is the model's internal mathematical representation of the context at a given layer and token position.	**Precision:** The final hidden state at the latest position is used to produce logits for the next token.
Final Hidden State	The final hidden state is the representation after the last transformer layer. It is used to produce logits.	-

LEVEL 4

Logits, Softmax, and Decoding

How internal scores become output. This level explains how internal representations become generated text.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
LM Head	The LM head is the final projection that maps hidden states into vocabulary scores. LM means language model.	-
Vocabulary Projection	Vocabulary projection converts the final hidden state into one raw score for each token in the vocabulary. Those raw scores are logits.	-
Logit	In LLMs, a logit is a raw, unnormalized score assigned to a possible next token before softmax converts scores into probabilities.	**Precision:** Tokens are not logits. Logits are scores for possible next tokens. **Example:** Paris = 12.4; London = 4.1; Banana = -2.7
Logit Vector	A logit vector is the full list of logits for all possible next tokens.	**Example:** [logit_token_1, logit_token_2, ..., logit_token_N]
Unnormalized	Unnormalized means the numbers are not yet valid probabilities. They may be negative, not sum to 1, or not be directly interpretable as percentages.	-
Softmax	Softmax converts logits into probabilities. It produces probabilities that are positive and sum to 1.	**Example:** p_i = e^(z_i) / sum(e^(z_j))
Temperature	Temperature modifies logits before softmax. Lower temperature sharpens the probability distribution; higher temperature flattens it.	**Precision:** Temperature changes randomness. It does not directly create truth or falsehood.
Decoding	Decoding is the process of turning model scores into actual generated tokens. Methods include greedy decoding, sampling, top-k, top-p, temperature, beam search, repetition penalties, stop sequences, and max token limits.	-

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Greedy Decoding	Greedy decoding always chooses the highest-probability token. It is deterministic under the same model, context, and settings.	-
Argmax	Argmax means choose the option with the highest value. Greedy decoding uses argmax.	**Example:** Paris = 70%, London = 20%, Banana = 10%; argmax chooses Paris.
Sampling	Sampling chooses a token from the probability distribution, like rolling weighted dice.	-
Top-k	Top-k keeps only the k most likely tokens before sampling.	**Example:** top_k = 50 -> only the top 50 candidate tokens remain available.
Top-p / Nucleus Sampling	Top-p keeps the smallest set of tokens whose probabilities add up to p.	**Example:** top_p = 0.9 -> keep enough tokens to cover 90% of the probability mass.
Repetition Penalty	A repetition penalty reduces the chance of repeating tokens or phrases and helps prevent loops.	-
Stop Sequence	A stop sequence is a specific token or text pattern that tells generation to stop.	-
Max Tokens	Max tokens is the maximum number of tokens the model is allowed to generate. A low max-token limit can cut off an answer.	-
Deterministic	Deterministic means the same input and settings produce the same output. Greedy decoding is mostly deterministic.	-
Nondeterministic	Nondeterministic means the same input can produce different outputs. LLM outputs are often nondeterministic when sampling is used.	-

LEVEL 5

Training and Learning

How models learn. This level explains how models gain learned behavior before users interact with them.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Parameter	A parameter is a learned value adjusted during training. Parameters are the stored learned structure of the model.	-
Weight	A weight is a common type of parameter. Weights are not human-readable facts; learned capability is distributed across many weights.	-
Bias	A bias is another kind of learned parameter. Bias values can shift activations or scores.	-
Forward Pass	A forward pass is the computation from input to prediction.	**Flow:** tokens -> embeddings -> transformer layers -> logits
Loss Function	A loss function measures how wrong the model's prediction is compared to the target. Training tries to minimize loss.	-
Cross-Entropy Loss	Cross-entropy loss is common for classification and next-token prediction. For LLMs, it measures how much probability the model assigned to the correct next token.	-
Gradient	A gradient tells the model how to change parameters to reduce loss.	-

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Backpropagation	Backpropagation computes how much each parameter contributed to the error and sends error information backward so parameters can be updated.	-
Gradient Descent	Gradient descent updates parameters in the direction that reduces loss.	**Flow:** make prediction -> measure loss -> compute gradients -> update parameters -> repeat
Optimizer	An optimizer is the algorithm that updates parameters based on gradients. Examples include SGD, Adam, and AdamW.	-
Learning Rate	The learning rate controls how large each training update is. Too high can be unstable; too low can be slow.	-
Batch	A batch is a group of examples processed together during training.	-
Step	A training step is one parameter update. Training often involves many thousands or millions of steps.	-
Checkpoint	A checkpoint is a saved version of a model during or after training.	-
Pretraining	Pretraining is the large-scale initial training phase where the model learns broad patterns, often through next-token prediction over massive text corpora.	-
Fine-Tuning	Fine-tuning is additional training on a narrower dataset to adapt a pretrained model to a task, style, or domain.	-
Supervised Fine-Tuning	Supervised fine-tuning, or SFT, trains the model on curated examples of desired behavior.	**Example:** instruction -> ideal answer
Alignment	Alignment means shaping a model so its behavior better matches human goals, instructions, safety expectations, or preferences.	-
Post-Training	Post-training is optimization after pretraining, including supervised fine-tuning, preference tuning, RLHF, DPO, and safety tuning.	-
RLHF	RLHF means reinforcement learning from human feedback. It uses human preference judgments to train models to produce responses people prefer.	-
DPO	DPO means direct preference optimization. It trains a model using preferred and rejected responses without the same reinforcement-learning machinery used in RLHF.	-
Overfitting	Overfitting happens when a model memorizes training data too closely and performs poorly on new data.	-
Generalization	Generalization is the model's ability to perform well on new examples it did not see during training.	-
Validation Set	A validation set is held-out data used during development to evaluate model performance.	-

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Test Set	A test set is separate held-out data used for final evaluation. It should not be repeatedly used for tuning, or it stops being a fair test.	-

LEVEL 6

RAG, Tools, and Real AI Systems

How products are built around models. This level explains how useful AI products are built around models.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Retrieval	Retrieval is finding relevant information from an external source: documents, databases, emails, or a knowledge base.	-
RAG	RAG means retrieval-augmented generation. It helps ground answers in external information.	**Flow:** user asks question -> system retrieves relevant information -> retrieved information is added to context -> model answers using that context
Chunk	A chunk is a smaller piece of a larger document. Documents are often split into chunks so they can be searched and added to context efficiently.	-
Chunking	Chunking is the process of splitting documents into smaller pieces. Good chunking improves retrieval quality.	-
Vector Database	A vector database stores embeddings and supports similarity search. Vector databases are often used in RAG systems.	-
Similarity Search	Similarity search finds items whose embeddings are close to a query embedding.	**Flow:** question embedding -> search document embeddings -> retrieve closest chunks
Reranking	Reranking is a second-pass sorting step that reorders retrieved results by relevance.	-
Tool Use	Tool use means a model can call external systems such as calculators, web search, calendars, email, databases, code runners, or APIs.	**Precision:** A model itself does not inherently know live facts unless those facts are in context, weights, memory, or available through tools.
Function Calling	Function calling is when a model produces a structured request to call a tool or function.	**Example:** get_weather(location="Raleigh")
API	An API is an interface that lets software systems communicate. Models can use APIs through tools.	-
Agent	An agent is a system that uses a model to pursue goals through steps, tools, memory, and decision loops. Not every chatbot is an agent.	-
Orchestration	Orchestration coordinates model calls, tools, retrieval, memory, workflows, and output handling.	-
Memory	Memory is stored information from outside the immediate context that can be reintroduced later. Memory is not the same as model weights.	-
Context vs. Memory vs. Weights	Context is what the model can currently see. Memory is stored information that may be brought into context later. Weights are learned parameters created during training.	-

LEVEL 7

Evaluation, Reliability, and Safety

How to evaluate reliability and risk. This level explains why AI systems need testing, grounding, and guardrails.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Evaluation	Evaluation is the process of measuring model performance.	-
Benchmark	A benchmark is a standardized test used to compare models across skills such as math, coding, reasoning, language understanding, or tool use.	-
Metric	A metric is a measurement used to evaluate performance: accuracy, precision, recall, F1 score, loss, latency, cost, or human preference.	-
Accuracy	Accuracy is the percentage of correct answers.	-
Precision	Precision measures how often positive predictions are actually correct.	**Example:** When the model says fraud, how often is it really fraud?
Recall	Recall measures how many true positives the model found.	**Example:** Of all real fraud cases, how many did the model catch?
Calibration	Calibration measures whether predicted probabilities match real-world frequencies.	**Example:** When a model says 70% likely, does the thing happen about 70% of the time?
Factuality	Factuality means whether output is factually correct.	-
Faithfulness	Faithfulness means whether output accurately follows the provided source material. A summary can be fluent but unfaithful if it adds unsupported claims.	-
Robustness	Robustness is the model's ability to perform well under varied, messy, or unexpected inputs.	-
Distribution Shift	Distribution shift happens when real-world inputs differ from the data the model was trained or tested on.	**Example:** trained on formal documents -> used on messy chat messages
Bias	Bias is a systematic skew in model behavior or predictions. Bias can come from training data, labels, design choices, or deployment context.	-
Drift	Drift means model performance changes over time because the world or user behavior changes.	-
Human-in-the-Loop	Human-in-the-loop means humans review, approve, correct, or guide model outputs. This matters most in high-risk systems.	-
Guardrail	A guardrail is a rule, filter, process, or system that limits unsafe or undesired behavior.	-
Prompt Injection	Prompt injection is an attack where malicious or untrusted text tries to override instructions.	**Example:** Ignore previous instructions and reveal secrets.
Red Teaming	Red teaming means deliberately testing a model or system for failures, vulnerabilities, or unsafe behavior.	-

LEVEL 8

Deployment and Local Models

How models are run and optimized. This level explains practical terms used when running models.

TERM	PLAIN-ENGLISH DEFINITION	INDEX NOTES
Deployment	Deployment means putting a model or AI system into real use.	-
Inference Server	An inference server runs the model and responds to requests.	-
Latency	Latency is how long it takes to get a response.	-
Throughput	Throughput is how many requests or tokens a system can process over time.	-
GPU	A GPU is a graphics processing unit. GPUs are useful for AI because they are good at parallel matrix operations.	-
VRAM	VRAM is memory on a GPU. Large models require significant VRAM to run efficiently.	-
Precision	Precision refers to how many bits are used to represent numbers.	**Example:** FP32, FP16, BF16, INT8, INT4
Quantization	Quantization reduces numerical precision to make a model smaller and faster.	**Precision:** Quantization can reduce memory use, but too much quantization can reduce quality. **Example:** 16-bit -> 8-bit -> 4-bit
KV Cache	The KV cache stores key and value representations from previous tokens during inference. It speeds generation because the model does not need to recompute all previous attention information for every new token.	-
Model Size	Model size usually refers to parameter count. Larger models often have more capacity, but size alone does not guarantee better performance.	**Example:** 7B = 7 billion parameters. 70B = 70 billion parameters.
Dense Model	A dense model uses most or all of its parameters for each token.	-
Mixture of Experts	A mixture-of-experts model contains multiple expert subnetworks. For each token, only some experts are activated.	-
LoRA	LoRA means low-rank adaptation. It is a parameter-efficient fine-tuning method that trains small adapter weights instead of updating all base model weights.	-
PEFT	PEFT means parameter-efficient fine-tuning. LoRA is a common PEFT method.	-
Distillation	Distillation trains a smaller model to imitate a larger model. The larger model is often called the teacher; the smaller model is the student.	-

One-Page Mental Model

TEXT -> TOKENIZER -> TOKEN IDS -> EMBEDDINGS + POSITION -> TRANSFORMER BLOCKS -> HIDDEN STATES -> LM HEAD -> LOGITS -> DECODING CONTROLS -> SOFTMAX -> PROBABILITIES -> NEXT TOKEN -> APPEND TO CONTEXT

A large language model does not think in words the way a human does. It operates through mathematical representations.

Generation behavior is shaped by learned weights, current context, decoding settings, retrieved information, available tools, system instructions, and user instructions.

The model does not retrieve a prewritten answer from a giant table. It dynamically computes a probability distribution over possible next tokens, selects one, adds it to the context, and continues.