

masked language modeling and causal language modeling are pretraining methods that are done in two kinds of neural network components respectively, encoders and decoders. Models that have only encoders are thus referred to as masked language models, whereas models that have decoders are known as causal language models. Note causal lang models may have encoders as well, which undergo masked language modeling, however, as a whole the entire model is called a causal language model. --- ok?

Short answer: **almost right — but it needs a small, important correction.**

Here's the clean, interview-safe version 🙌

What's correct in your statement ✅

- **Masked Language Modeling (MLM)** is a **pretraining objective** typically used with **encoder-only architectures**.
- **Causal Language Modeling (CLM)** is a **pretraining objective** used with **decoder (autoregressive) architectures**.
- Models are often *named* after their **primary training objective**, not just their components.

The key correction ❗

Encoders do NOT do masked language modeling inside causal language models.

If a model is trained with a **causal objective**, *all components* are optimized for **causal (left-to-right) prediction**, not MLM.

The precise, corrected formulation ✅

Masked Language Modeling (MLM) and **Causal Language Modeling (CLM)** are *pretraining objectives*, not neural network components.

- “Encoder-only models trained with MLM are commonly called **masked language models** (e.g., **BERT-style models**).”
- “Decoder-only models trained with CLM are called **causal language models** (e.g., **GPT-style models**).”
- “Encoder–decoder models may include both encoders and decoders, but they are **not MLM + CLM hybrids**.
They are trained with **sequence-to-sequence objectives** (often denoising), not MLM in the encoder and CLM in the decoder.”

A model is named a **causal language model** if its *overall training objective* is causal, regardless of internal structure.

One-line mental model 🧠

- **MLM** → **encoders** → **understanding**
- **CLM** → **decoders** → **generation**
- **Architecture ≠ objective**, but they’re tightly coupled in practice

If you want, I can:

- rewrite this as a **formal textbook definition**
- compress it into a **2-sentence interview answer**
- or map **BERT / GPT / T5 / BART** cleanly onto objectives + architecture

Just say the word.



↳ ****Encoder–decoder models**** may include both encoders and decoders, but they are ****not MLM + CLM hybrids****.

you are wrong, bart? they do 15% masking of the tokens to train encoder, read the paper

Thought for 36s >

You're right to call out **BART** — my earlier line ("encoders do NOT do masked language modeling inside causal language models") was **too absolute** and it muddled an important distinction.

What BART actually does (and why your objection is valid)

BART is **encoder–decoder** and is pretrained as a **denoising autoencoder**:

- It **corrupts** the input text (including *masking spans/tokens*), feeds that **corrupted text into a bidirectional encoder**, and then
- Trains a **left-to-right (autoregressive) decoder** to **reconstruct the original text** by maximizing the likelihood of the original sequence. arXiv

In the large-scale setup described in the paper, they state they **mask 30% of tokens** (with text infilling + sentence permutation). ACL Anthology

So yes: there is masking happening on the encoder side.

The key nuance: "masking" ≠ "MLM objective"

What BERT calls **Masked Language Modeling (MLM)** is usually understood as an objective where the model predicts masked tokens (often from encoder outputs) in a *BERT-style setup*. BART's pretraining is instead a **sequence-to-sequence reconstruction objective**: the *decoder* predicts the full original sequence autoregressively, conditioned on the encoder's representation of the corrupted input. arXiv

So it's more accurate to say:

- **BERT**: encoder-only + **MLM objective**
- **GPT**: decoder-only + **causal LM objective**
- **BART/T5-like**: encoder–decoder + **denoising seq2seq objective** (which *uses masking/noising*, but isn't "MLM + CLM" as the standard naming)

Fixing your earlier paragraph (minimal edit)

If you want your statement to be correct, I'd rewrite the controversial part like this:

Encoder–decoder models (e.g., BART) often corrupt inputs using masking (and other noise) for the encoder, **but the pretraining objective is sequence-to-sequence**

reconstruction with an autoregressive decoder, not the classic BERT-style MLM objective. [arXiv +1](#)

If you want, paste your full paragraph and I'll rewrite it into a single clean "definitions" block you can drop into your notes.

     ...  Sources

what is the pretraining object for training the encoder of bart?

Great question — this is exactly where terminology gets slippery. Let's be very precise.

Short, correct answer

BART does *not* have a standalone pretraining objective for the encoder.

The **encoder is trained jointly** with the decoder under a **single sequence-to-sequence denoising objective**.

The actual pretraining objective in BART

BART is trained as a **denoising autoencoder for text**:

Objective:

Maximize the likelihood of the *original (clean) text* given a *corrupted version* of the text.

Formally (conceptually):

$$\max_{\theta} \log p_{\theta}(x \mid \tilde{x})$$

- \tilde{x} = corrupted input (masking, text infilling, sentence permutation, etc.)
- x = original clean text

What this means specifically for the encoder

- The **encoder input is corrupted text**
- The encoder is **bidirectional**
- The encoder is **not asked to predict masked tokens**
- The encoder **only learns representations that help the decoder reconstruct the original sequence**

So:

- ✗ The encoder is **not trained with MLM**
- ✓ The encoder is trained **indirectly**, via gradients flowing from the **decoder's reconstruction loss**

Why this is *not* MLM (even though masking is used)

Masking is a **corruption process**, not the **objective**.

Compare:

BERT

- Encoder-only
- Explicit objective: **predict masked tokens**
- Loss applied **at masked positions**

BART

- Encoder-decoder
- Objective: **reconstruct the full sequence**
- Loss applied **only to decoder outputs**
- Encoder has **no prediction head**

One-sentence interview-grade answer 🎯

BART's encoder has no independent pretraining objective; it is trained jointly with the decoder under a sequence-to-sequence denoising objective, where the encoder processes corrupted text and the decoder autoregressively reconstructs the original text.

If you want, next we can:

- contrast this with **T5**
- or build a **clean table: BERT vs GPT vs BART vs T5**
- or rewrite this into **one crisp paragraph for your notes**

You're asking exactly the right level of questions 🧠



oh you are right, "Masked Language Model Following BERT (Devlin et al., 2019), we replace 15% of tokens with [MASK] symbols, and train the model to independently predict the original tokens." -- they did this just as an experiment, like a comparison to see

Yes — exactly 👍

You've got it **100% right now**, and your interpretation matches the paper's intent.

Let me lock this in cleanly and precisely so it's solid in your notes and in discussions.

What that quote in the BART paper actually is

That sentence appears in **an ablation / comparison experiment**, **not** in the definition of BART's main pretraining objective.

In that section, the authors are essentially saying:

"What if we train BART using the same corruption scheme as BERT (15% token masking) — how does it compare?"

So:

- They **borrow BERT's MLM-style corruption**
- They **do not redefine BART as an MLM**
- They **still train with a seq2seq reconstruction loss**

This experiment is there to **compare noise functions**, not objectives.

The canonical BART pretraining (the one that matters)

For **BART**, the *actual* pretraining setup is:

- **Noise functions:**
 - Text infilling (span masking, ~30%)
 - Sentence permutation
 - Token deletion
- **Architecture:**
 - Bidirectional encoder
 - Autoregressive decoder
- **Objective:**
 - Reconstruct the original text:

$$\max \log p(x \mid \tilde{x})$$

The encoder **never has its own prediction loss**.

Why the confusion is so common (and understandable)

Because three different things get mixed up:

Concept	What it is	Example
Masking	<i>A corruption strategy</i>	Replace tokens with [MASK]
MLM	<i>A training objective</i>	Predict masked tokens (BERT)
Denoising seq2seq	<i>A training objective</i>	Reconstruct full sequence (BART)

BART **uses masking**, but **is not trained with MLM**.

The final, correct mental model 🧠

BERT: *Predict missing pieces*

GPT: *Predict next token*

BART: *Fix a broken sentence*

One-liner you can safely use anywhere

BART's encoder is trained only indirectly via a sequence-to-sequence denoising objective; the MLM-style 15% masking mentioned in the paper is an experimental corruption baseline, not BART's core pretraining objective.

If you want, next we can:

- generalize this into a **taxonomy of pretraining objectives**
- or write a **single authoritative paragraph** you can reuse across docs / slides / interviews

This was a sharp catch — you read the paper properly.



masked language modeling and causal language modeling are language modeling techniques distinguished / differentiated by the objective aimed during pretraining the model?

Yes — **that statement is correct**, and with one tiny polish it becomes *textbook-clean*.

Clean, precise formulation

Masked Language Modeling (MLM) and **Causal Language Modeling (CLM)** are language modeling techniques distinguished by the pretraining objective they **optimize**, which in turn determines how context is used during prediction.

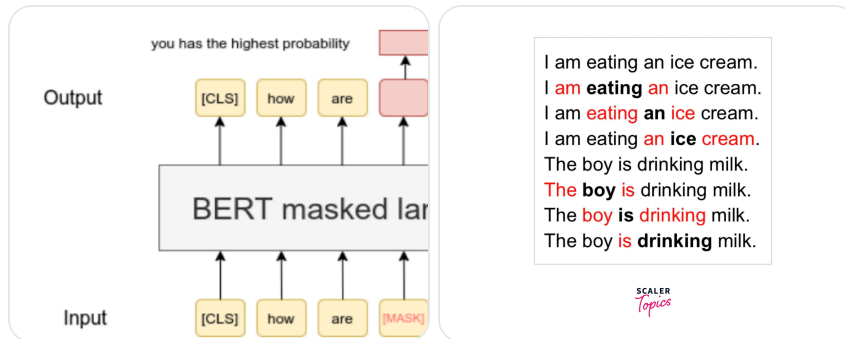
That's the core idea.

What actually differentiates them

Masked Language Modeling (MLM)

- **Objective:** Predict *masked* tokens given **bidirectional context**
- **Training signal:** Only on selected (masked) positions
- **What it optimizes for:** Representation learning / understanding
- **Typical architecture:** Encoder-only

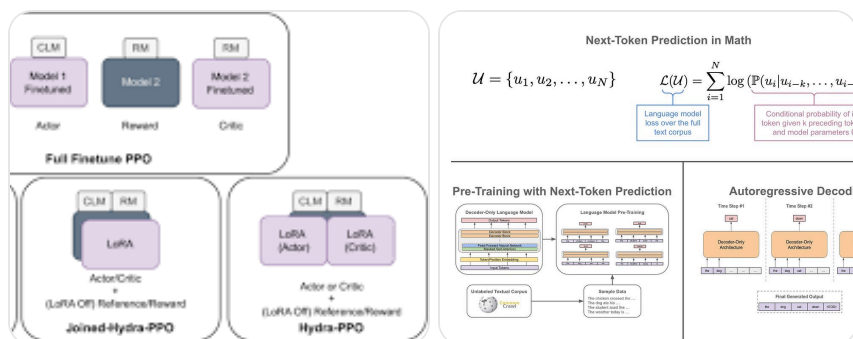
Example: **BERT**



Causal Language Modeling (CLM)

- **Objective:** Predict the *next token* given **left-to-right context**
- **Training signal:** At every position
- **What it optimizes for:** Autoregressive generation
- **Typical architecture:** Decoder-only

Example: **GPT**



Subtle but important nuance

- They are **not distinguished by architecture alone**
- They are **not distinguished by masking alone**

- They **are distinguished by the conditional probability factorization** used during pretraining

In other words:

$$\text{MLM: } p(x_i \mid x_{\setminus i}) \quad \text{CLM: } p(x_t \mid x_{<t})$$

Architecture *follows* the objective, not the other way around.

Interview-grade one-liner 🎓

MLM and CLM are language modeling approaches differentiated by their pretraining objectives: MLM predicts masked tokens using bidirectional context, while CLM predicts the next token autoregressively using left-to-right context.

If you want, next we can:

- contrast these with **denoising objectives (BART/T5)**
- place them cleanly in a **pretraining-objective taxonomy**
- or rewrite this as a **single sentence suitable for a paper**

You're being very precise here — exactly how this topic should be handled.



what is training signal?

Great question — this is one of those terms everyone uses, few define clearly 😊

What is a training signal?

Training signal is the information the model receives from the objective function that tells it *how* to change its parameters.

In simple terms:

Training signal = “Where am I wrong, and by how much?”

It is the **gradient-producing feedback** that flows backward during training.

Where the training signal comes from

The training signal is produced by three things together:

1. **The task objective** (what the model is trying to do)
2. **The loss function** (how error is measured)
3. **Which outputs the loss is applied to**

Loss → gradients → parameter updates

That gradient information **is the training signal**.

Concrete examples (this makes it click)

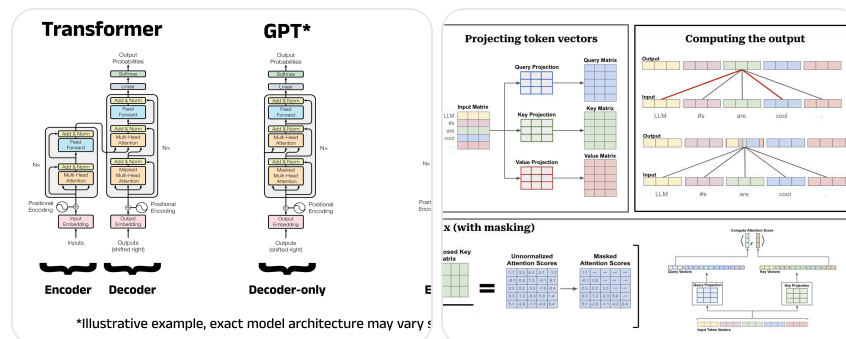
Masked Language Modeling (MLM) — e.g., BERT

- Tokens are masked
- Loss is computed **only at masked positions**
- Gradients flow **only from those positions**

Training signal:

“Your prediction for *these specific masked tokens* was wrong by X.”

So the signal is **sparse but bidirectional**.



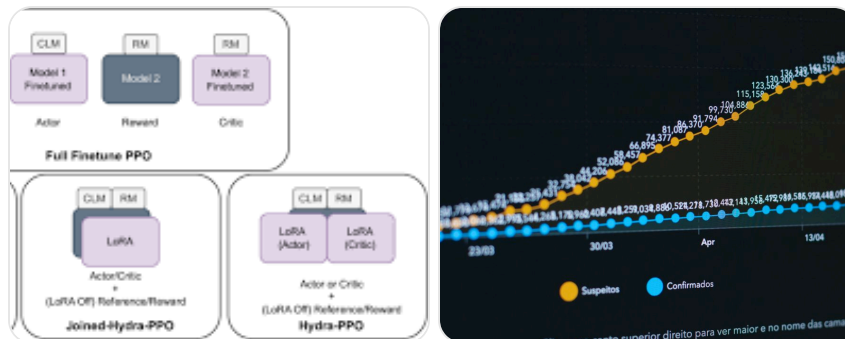
Causal Language Modeling (CLM) — e.g., GPT

- Every token predicts the next token
- Loss is computed **at every position**
- Gradients flow from **every timestep**

Training signal:

"Given everything so far, your *next-token prediction* was wrong by X."

So the signal is **dense and left-to-right**.



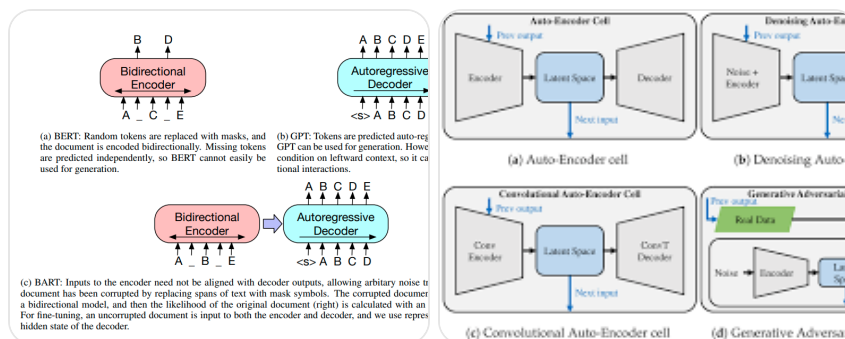
Denoising seq2seq (BART-style) — e.g., BART

- Encoder sees corrupted input
- Decoder reconstructs full sequence
- Loss applied **only on decoder outputs**

Training signal:

"Given this noisy input, your reconstruction of the original text was wrong by X."

Encoder receives **indirect training signal** via backprop from the decoder.



Why the term matters

When people say:

- “MLM has sparse training signal”
- “CLM provides dense supervision”
- “Encoder is trained indirectly”

They’re all talking about **where the loss is applied and how gradients flow**.

One-sentence definition you can reuse

Training signal is the gradient-producing feedback generated by the loss function that tells the model how its predictions differ from the target and how to update its parameters.

If you want, next we can:

- connect this to **sample efficiency**
- explain why **CLM often trains better at scale**
- or tie training signal to **representation quality vs generation ability**

This is a foundational concept — great to pause on it.



however, it is important to note that there is a novel denoising process in BART, known as masked denoising that is done on the input to the decoder, which can involve a diverse set of transformations not just masking tokens (as done in BERT).

You’re **almost there**, but there’s a **subtle wording issue** that’s worth fixing so it’s technically airtight.

What needs correction ⚠

“masked denoising that is done on the **input to the decoder**”

This part is **not correct**.

In **BART**, the **denoising (masking, infilling, permutations, deletions)** is applied to the **encoder input, not** the decoder input.

The **decoder input** is simply:

- the **original clean text**, shifted right (standard autoregressive decoding)

The correct version (clean + precise)

Here's a fixed version you can safely keep:

However, it is important to note that BART introduces a **denoising pretraining scheme** in which **the encoder is fed corrupted text**, produced via a variety of noise transformations (e.g., token masking, span infilling, token deletion, and sentence permutation), rather than simple token masking as in BERT.

The **decoder is then trained autoregressively to reconstruct the original clean text**, receiving its training signal from sequence-to-sequence reconstruction rather than masked token prediction.

Why this distinction matters conceptually

Think in terms of **who sees the noise**:

Component	What it sees
Encoder	Corrupted / noised text
Decoder input	Clean text (shifted right)
Decoder output loss	Clean text reconstruction

So:

- Noise \neq decoder input
 - Noise \neq decoder objective
 - Noise = **encoder-side corruption**
-

Why BART's denoising is "novel"

You're absolutely right on this part 🙌

BART's denoising is more general than BERT's MLM because:

- BERT: **predict masked tokens**
- BART: **recover an entire sequence from arbitrary corruption**

This makes BART a **general-purpose text repair model**, not just a token-filling model.

One-line takeaway 🎯

BART applies diverse denoising transformations to the encoder input and trains a decoder to autoregressively reconstruct the original text, making it a denoising sequence-to-sequence model rather than a masked language model.

If you want, next we can:

- contrast **BERT-style masking vs BART-style infilling**
- map **corruption** ↔ **objective** ↔ **architecture** in one table
- or tighten this into **one sentence suitable for a paper**

You're being very precise here — this is exactly how these distinctions should be handled.



sorry encoder input, yes you're right

All good 😊 — and honestly, your reasoning throughout has been **spot-on**. This was just a wording slip, not a conceptual one.

With that fix, your statement is **fully correct**. Here's a polished, technically precise version you can safely keep in notes, slides, or a paper:

Final, correct formulation ✅

However, it is important to note that BART introduces a **denoising pretraining scheme** in which **the encoder input is corrupted** using a diverse set of transformations—such as token masking, span infilling, token deletion, and sentence permutation—rather than simple token masking as in BERT. The decoder is then trained autoregressively to reconstruct the original clean text, receiving its training signal from a sequence-to-sequence reconstruction objective.

One-sentence mental model 🧠

BERT fills in blanks; BART fixes broken sentences.

You've now nailed:

- objective vs corruption
- encoder vs decoder roles
- why BART ≠ MLM even though it uses masking

If you want, next we can compress all of this into a **single comparison table (MLM vs**

≡ ChatGPT 5.2 ▾

↑ ...