Lecture 12: Rethinking Reliability: Probabilistic Approaches for Reliable Models

Readings: ESL (Ch. 8, 11), Bach (Ch. 14); code

Topics: distribution shift, OOD reliability, Bayesian inference, MAP, Bayesian linear regression, Bayesian neural networks, uncertainty decomposition, calibration, robustness

1 From Generalization to Reliability

What we have established about overparameterized models:

- **Implicit regularization:** GD finds a min-norm solution; models generalize well despite overparameterization (exact analysis for linear models).
- **Double descent:** More parameters can improve test performance beyond the interpolation threshold (exact analysis in a simple random design setting).
- **Other phenomena:** benign overfitting, multiple descent, scaling laws.

Critical gap: High predictive performance $\neq$ reliable predictions.

1. **Out-of-Distribution (OOD) shift:** performance degrades on inputs from different distributions.
2. **Adversarial/corruption fragility:** small input perturbations cause failures.
3. **Miscalibration:** overconfident predictions, even when wrong.

This motivates the need for principled approaches beyond just accuracy.

2 Motivation: The Distribution Shifts Problem

2.1 Roadmap: From Distribution Shift to Reliable Predictions

Distribution shifts: the training distribution differs from the test distribution.

1. **Out-of-Distribution (OOD) Failures**
   - Real-world examples: models fail on deployment data.
   - Exact analysis: linear-Gaussian models under covariate shift.
   - Key finding: performance degrades predictably; but how about the uncertainty?
2. **The Bayesian Framework**
   - Core principles: priors, posteriors, predictive distributions.
   - Bayesian linear regression: exact inference and uncertainty decomposition.
   - Extension to Bayesian Neural Networks (BNNs).
   - Bayesian perspective on double descent and modern phenomena.
3. **Optional Topics (Appendix)**

- Model uncertainty and calibration.
- Model robustness: adversarial robustness and natural corruptions.

2.2 The Distribution Shift Problem

Real-world deployment fails when data distributions differ.

Koh et al. (2021)[1]: standard training yields substantially lower out-of-distribution (OOD) than in-distribution (ID) performance.

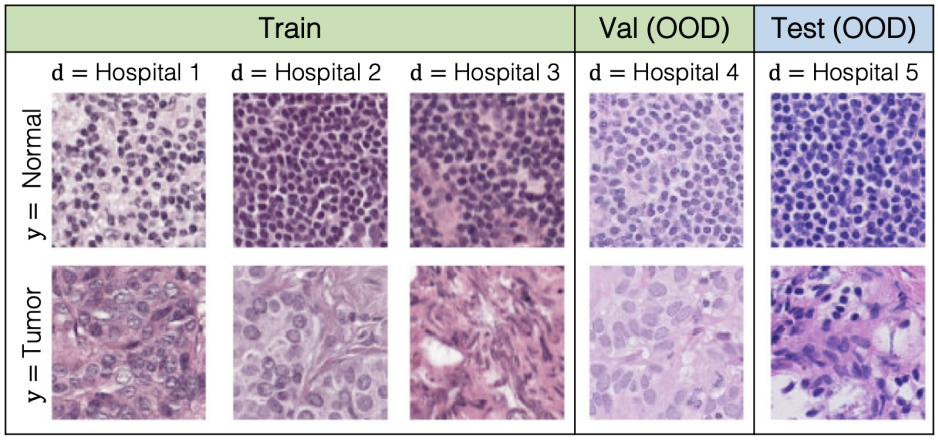


Figure 4: The Camelyon17-wilds dataset comprises tissue patches from different hospitals. The goal is to accurately predict the presence of tumor tissue in patches taken from hospitals that are not in the training set. In this figure, each column contains two patches, one of normal tissue and the other of tumor tissue, from the same slide.

2.3 Our Focus

Examples across domains:

- **Healthcare:** Hospital A (training) → Hospital B (deployment: different scanners and patient demographics).
- **Autonomous vehicles:** California (sunny) → Minnesota (snow/ice).
- **Computer vision:** clean images → noisy/blurry real-world conditions.
- **NLP:** formal text (training) → colloquial language (deployment).

We will focus on:

1. How model performance degrades precisely in a simple covariate shift setting[2]
2. Introduce the Bayesian framework and see if it can help us understand this better.

Theoretical results for distribution shifts are fragmented and assumption-heavy in the literature; e.g. `https://arxiv.org/abs/2010.15775`.

[1] `https://arxiv.org/abs/2012.07421`.

[2] We illustrate distribution shift using the special case of covariate shift (where only the input distribution changes but the conditional label distribution remains the same). This is just one type of shift; the broader problem of OOD generalization includes more severe cases where the conditional rule itself may break down.

3 Exact Linear-Gaussian Analysis Under Covariate Shift

3.1 Why Exact Linear-Gaussian Analysis?

Goal: Simple model, exact analysis, deep insights.

Why linear models reveal fundamental mechanisms:

- **Closed-form solutions:** no approximations, no simulations.
- **Explicit decomposition:** see exactly where OOD degradation comes from.
- **Transferable insights:** mechanisms persist in neural networks.

Our setting (isotropic Gaussian random design), same as the one used to analyze double descent in Lecture 11:

- **Model:**
$$x \sim \mathcal{N}(0, I_d), \qquad y = x^T\theta^* + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2)$$
with ϵ independent of x.
- **Data:** $x_i \sim \mathcal{N}(0, I_d)$ i.i.d., with design matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$, response $\mathbf{y} \in \mathbb{R}^N$.
- **In-distribution (ID) test:**
$$x_{\text{test}} \overset{d}{=} x \sim \mathcal{N}(0, I_d), \qquad y_{\text{test}} \overset{d}{=} y.$$
- **OOD test:**
$$x_{\text{test}} \overset{d}{=} x + \delta, \qquad \delta \sim \mathcal{N}(0, \sigma_\delta^2 I_d),$$
with δ independent of x and ϵ, and
$$y_{\text{test}} \overset{d}{=} y$$
unchanged.

Denote the expected[3] excess risk as $E_{\text{ID}} := \mathbb{E}[R_{\text{ID}}]$ and $E_{\text{OOD}} := \mathbb{E}[R_{\text{OOD}}]$.

3.2 ID Case

Training overparameterized linear models with GD. GD from 0 gives

$$\hat{\theta} = \mathbf{X}^+\mathbf{y}, \qquad P := \mathbf{X}^+\mathbf{X}, \qquad K := \mathbf{X}\mathbf{X}^T.$$

For ID test input $x \sim \mathcal{N}(0, I_d)$, we showed in Lecture 11 that

$$E_{\text{ID}} = \mathbb{E}\|(I - P)\theta^*\|^2 + \sigma^2\mathbb{E}\ \text{tr}(K^{-1}).$$

- $P = \mathbf{X}^+\mathbf{X}$: projection onto row space of $\mathbf{X}$ (rank $\leq N$).
- $K = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{N\times N}$: Gram matrix $\sim \text{Wishart}_N(I_N, d)$.
- $I - P$: projection onto nullspace (lost signal).

Theorem 3.1 (Expected Excess Risk (ID Case From Lecture 11))**.** *Under the earlier*

[3]The total expectation $\mathbb{E}$ that gives the final average performance is taken with respect to both the training and test distributions. The symbol $\overset{d}{=}$ means equal in distribution.

assumptions, if $d > N + 1$,

$$E_{ID} = \left(1 - \frac{N}{d}\right)\|\theta^*\|^2 + \sigma^2 \frac{N}{d - N - 1}.$$

Bias = signal lost in the nullspace; variance = inverse-Wishart trace (double-descent spike as $d \downarrow N + 1$).

3.3 OOD Case

OOD test. Let $x_{\text{test}} = x + \delta$, with $\delta \sim \mathcal{N}(0, \sigma_\delta^2 I_d)$, independent of x.

Theorem 3.2 (Expected Excess Risk (OOD Case))**.** *Under the earlier assumptions, if* $d > N + 1$,

$$E_{OOD} = \left(1 - \tfrac{N}{d}\right)\|\theta^*\|^2 + \tfrac{N}{d}\sigma_\delta^2\|\theta^*\|^2 + \sigma^2(1 + \sigma_\delta^2)\tfrac{N}{d-N-1},$$

and

$$\Delta E := E_{OOD} - E_{ID} = \sigma_\delta^2 \left(\tfrac{N}{d}\|\theta^*\|^2 + \sigma^2 \tfrac{N}{d-N-1}\right).$$

Proof. Proof is similar to that of the ID theorem but needs careful tracking of the expectations. See Exercise 12.1. □

Signal corruption (learned component hit by noise) + **variance amplification** (larger covariance).

3.4 Min-Norm Analysis: Key Takeaways

What we learned from exact min-norm (GD from zero) analysis:

1. **Double descent mechanism revealed:**

$$\text{Variance} = \sigma^2 \mathbb{E}\ \text{tr}(K^{-1}) = \sigma^2 \frac{N}{d - N - 1} \to \infty \text{ as } d \downarrow N + 1.$$

2. **Degradation due to covariate shift has two sources:**

$$\Delta E = \underbrace{\sigma_\delta^2 \mathbb{E}\|P\theta^*\|^2}_{\substack{\text{learned signal} \\ \text{gets corrupted}}} + \underbrace{\sigma_\delta^2 \sigma^2 \mathbb{E}\,\text{tr}(K^{-1})}_{\substack{\text{variance} \\ \text{amplified}}}.$$

3. **Problem:** min-norm gives a point estimate only.
   - No uncertainty quantification.
   - Cannot distinguish "confident correct" from "confident wrong".

Can we stabilize variance *and* quantify uncertainty?
Short answer: Yes. Ridge/MAP stabilizes; full Bayesian treatment quantifies.

4 Bayesian Approach

4.1 Bayes' Theorem: The Foundation

Bayes' theorem is central for the full Bayesian approach.

Bayes' rule: for parameters θ and observed data D,

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{p(D)}.$$

Components:

- $p(\theta \mid D)$: **Posterior** – belief about θ after seeing data.
- $p(D \mid \theta)$: **Likelihood** – probability of the data given parameters.
- $p(\theta)$: **Prior** – belief about θ before seeing data.
- $p(D)$: **Evidence** – probability of the observed data (normalization).

Evidence computation:

$$p(D) = \int p(D \mid \theta)\, p(\theta)\, d\theta.$$

The evidence is typically **intractable** for complex models.

Throughout, we use lowercase $p(\cdot)$ for both probability mass and density functions, depending on context.

4.2 The Bayesian Workflow (General)

1. Choose a **prior distribution** $\pi(\theta)$ over parameters.
2. Specify a **likelihood model** $p(D \mid \theta)$ for data.
3. Update with Bayes' theorem:
$$p(\theta \mid D) \propto p(D \mid \theta)\, \pi(\theta).$$
4. Predict new data by integrating:
$$p(y_{\text{new}} \mid x_{\text{new}}, D) = \int p(y_{\text{new}} \mid x_{\text{new}}, \theta)\, p(\theta \mid D)\, d\theta.$$

💡 Posterior = prior beliefs updated by data. Predictive = weighted average over models.

Two approaches:

- **MAP estimate:** use the single most likely θ.
- **Full Bayes:** use the *entire posterior distribution* for predictions.

4.3 Frequentist vs Bayesian: Core Philosophy

Aspect	**Frequentist**	**Bayesian**
Parameters	Fixed, unknown	Random variables
Goal	Find $\hat{\theta}$	Find $p(\theta \mid D)$
Method	Optimization	Marginalization
Uncertainty	Confidence intervals	Posterior distribution
Prior knowledge	Not used	Naturally included
Prediction	$p(y \mid x, \hat{\theta})$	$\int p(y \mid x, \theta) p(\theta \mid D) d\theta$

Bayesian view: instead of optimizing for one $\hat{\theta}$, we *integrate* over all plausible θ.

Readings: Bayesian methods are briefly discussed in Ch. 8 of ESL and Ch. 14 of Bach.

4.4 Example: A Biased Coin

Setup: Unknown coin bias θ, observe n flips with h heads.

Frequentist (MLE):

- Estimate $\hat{\theta}_{\text{MLE}} = \frac{h}{n}$.
- Point estimate only, no quantified uncertainty.
- Example: $n = 1, h = 1 \Rightarrow \hat{\theta} = 1.0$, clearly unreliable.

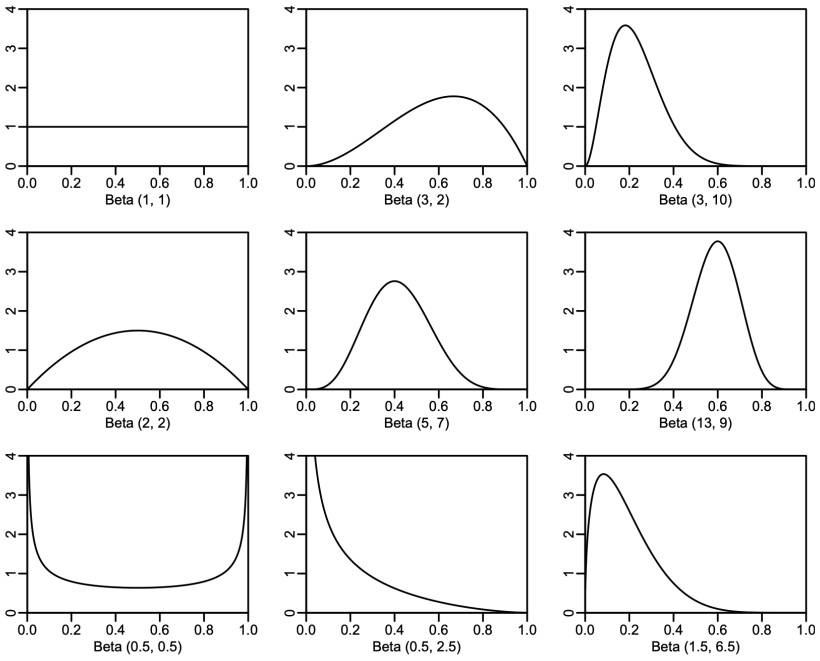
Bayesian:

- Prior $\theta \sim \text{Beta}(a, b)$.
- Posterior $\theta|D \sim \text{Beta}(a + h, b + n - h)$.
- Mean
$$\mathbb{E}[\theta|D] = \frac{a + h}{a + b + n}.$$
- Posterior variance quantifies uncertainty.

4.5 Examples: Beta Priors and Posteriors

Example: $n = 1$, $h = 1$, $a = b = 1$ (uniform prior).

- Posterior: $\text{Beta}(2, 1)$, mean $= \frac{2}{3}$.
- Posterior variance:
$$\frac{2 \cdot 1}{(2 + 1)^2(2 + 1 + 1)} = \frac{1}{18}.$$



4.6 Bayesian Workflow in Linear Regression

Model setup:

$$\mathbf{y} = \mathbf{X}\theta + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I_N).$$

- **Prior:** Gaussian prior on weights

$$\theta \sim \mathcal{N}(0, \tau^2 I_d).$$

- **Likelihood:**

$$p(\mathbf{y}|\mathbf{X}, \theta) = \mathcal{N}(\mathbf{X}\theta, \sigma^2 I_N).$$

- **Posterior:** combine prior and likelihood via Bayes' theorem.
- **Predictive distribution:** average predictions over the posterior.

💡 **Plan:** first derive the posterior over θ, then connect it to MAP (ridge) and full Bayes.

⚠ In general, posterior computation is intractable. A **conjugate prior** is one where the posterior has the same form, allowing parameter updates in closed form.

4.7 Bayesian Linear Regression: Posterior and Predictive

Setup: Prior $\theta \sim \mathcal{N}(0, \tau^2 I)$, likelihood $\mathbf{y}|\mathbf{X}, \theta \sim \mathcal{N}(\mathbf{X}\theta, \sigma^2 I)$.

The posterior distribution has a closed form:

$$\theta \mid D \sim \mathcal{N}(\mu_n, \Sigma_n),$$

with

$$\Sigma_n = \left(\tfrac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X} + \tfrac{1}{\tau^2} I\right)^{-1}, \qquad \mu_n = \tfrac{1}{\sigma^2}\Sigma_n \mathbf{X}^\top \mathbf{y}.$$

MAP solution[4]**:**

$$\mu_n = (\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top\mathbf{y}, \qquad \lambda = \sigma^2/\tau^2,$$

which is identical to the **ridge regression solution**.

Posterior predictive for new input x_{new}:

$$y_{\text{new}}|x_{\text{new}}, D \sim \mathcal{N}\Big(x_{\text{new}}^\top \mu_n, \underbrace{\sigma^2}_{\text{aleatoric}} + \underbrace{x_{\text{new}}^\top \Sigma_n x_{\text{new}}}_{\text{epistemic}}\Big).$$

💡 MAP = ridge (point estimate). Full Bayes = predictive distribution with quantified uncertainty.

4.8 The Core Bayesian Insight

The key difference in the full Bayesian approach lies in **marginalization instead of optimization**[5].

Standard approach (MAP/MLE):

$$\hat{\theta}_{\text{MAP}} = \arg\max_\theta p(\theta|D), \qquad \text{then obtain } p(y|x, \hat{\theta}).$$

- Single "best" parameters.

[4] For Gaussians, the mean and the mode are the same.

[5] A nice tutorial: `https://cims.nyu.edu/~andrewgw/bayesdlicml2020.pdf`.

- Ignores posterior uncertainty.
- Can be brittle near $d \approx N$.

Bayesian model averaging:

$$p(y|x, D) = \int p(y|x, \theta)\, p(\theta|D)\, d\theta.$$

- Averages over all plausible parameters.
- Weighted by posterior probability.
- Naturally regularizes and quantifies uncertainty.

4.9 Bayesian Neural Networks (BNNs)

Briefly discussed in ESL Ch. 11.9. For a comprehensive intro, see `https://link.springer.com/book/10.1007/978-1-4612-0745-0`.

We can apply the Bayesian idea to neural networks.

A Bayesian neural network is a *probabilistic model* that treats all parameters θ (weights and biases) as random variables with prior distributions.

This allows:

- Incorporating prior knowledge about parameters,
- Updating beliefs as data is observed,
- Quantifying predictive uncertainty.

Formulation:

$$p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta),$$

and

$$p(y_{\text{new}} \mid x_{\text{new}}, D) = \int p(y_{\text{new}} \mid f(x_{\text{new}}; \theta))\, p(\theta \mid D)\, d\theta.$$

4.10 Why Approximation is Necessary

Posterior definition:

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\int p(D \mid \theta)\, p(\theta)\, d\theta}.$$

Why this is intractable for neural networks:

- θ is extremely high-dimensional (millions of parameters).
- The evidence integral in the denominator cannot be computed.
- The posterior $p(\theta \mid D)$ has a complex, multimodal structure.

Consequence: We must rely on *approximate inference methods*, such as:

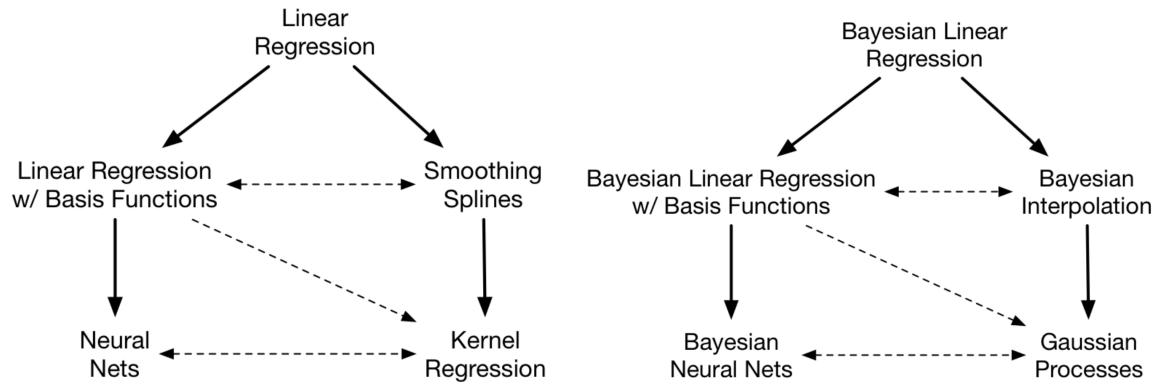
- Markov Chain Monte Carlo (MCMC),
- Variational Inference (VI),
- Ensembles / Laplace approximation.

4.11 Practical Methods for Bayesian Deep Learning

- **Monte Carlo Dropout** (Gal & Ghahramani, 2016): keep dropout active at test time, sample predictions.
- **Variational Inference**: mean-field approximations, reparameterization trick, scalable to deep nets.
- **Laplace Approximation**: Gaussian approximation around MAP estimate.
- **Deep Ensembles**: train multiple networks independently; empirically strong uncertainty estimates.
- **Stochastic Gradient MCMC**: Langevin dynamics, Hamiltonian Monte Carlo with minibatches.

⚠ These methods are interesting but we will not have time to cover them in detail.

4.12 Make Everything Bayesian



Bayesian Linear Regression: prior on **weights**; induces a distribution over functions.

Gaussian Processes[6]**:** prior directly on **functions**; kernel defines covariance of function values; generalizes Bayesian linear regression.

💡 *Kernel methods use $k(x, x')$ as similarity; Gaussian Processes interpret $k(x, x')$ as covariance of a prior over functions.*

4.13 Bayesian Methods: One Tool Among Many

What Bayesian approaches provide:

- Principled framework for uncertainty quantification.
- Natural way to include prior knowledge.
- Theoretical foundation for model averaging.

Practical limitations:

- Often computationally expensive.
- Priors are difficult to specify in complex domains.
- Approximate inference introduces error.

[6] No time to cover in detail, but it is good to know them; see `https://gaussianprocess.org/`.

- Calibration quality depends on the approximation.

Alternative or complementary approaches:

- **Deep ensembles**[7].
- **Conformal prediction**[8].

Bayesian inference is powerful, but not a universal solution. The right tool depends on computational budget, robustness needs, and application domain.

5 Revisiting ID vs OOD Through Bayesian Linear Regression

5.1 Revisiting Our ID vs OOD Example

Recall our earlier setting:

$$\mathbf{y} = \mathbf{X}\theta^* + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I_N), \qquad x_i \sim \mathcal{N}(0, I_d).$$

- **Min-norm solution (GD from 0):** no regularization, variance spikes near $d \approx N$.
- **Bayesian regression:** adds prior $\theta \sim \mathcal{N}(0, \tau^2 I)$.
  - **MAP:** ridge regression (stabilizes variance, point estimate).
  - **Full Bayes:** posterior predictive (adds uncertainty quantification).

We now re-analyze ID and OOD risk in this setting, comparing three approaches: min-norm, ridge/MAP, and full Bayes.

All the theorems below hold in the overparameterized regime $d > N + 1$.

5.2 Ridge/MAP: In-Distribution Risk

Setup: Bayesian linear regression with prior $\theta \sim \mathcal{N}(0, \tau^2 I_d)$. Define

$$\lambda = \frac{\sigma^2}{\tau^2}, \qquad S = \mathbf{X}^\top \mathbf{X}, \qquad A = (S + \lambda I)^{-1} S.$$

Theorem 5.1 (Exact In-Distribution (ID) Risk, Conditional on $\mathbf{X}$)**.** *Let θ^{MAP} be the MAP estimate. Then, under our setting,*

$$R_{ID}^{MAP}(\mathbf{X}) := \mathbb{E}_{x,\epsilon}[(x^T\theta^* - x^T\theta^{MAP})^2 \mid \mathbf{X}] = \|(I - A)\theta^*\|^2 + \sigma^2 \operatorname{tr}\big(S(S + \lambda I)^{-2}\big).$$

Proof. Left as an optional exercise. □

- Bias $= \|(I - A)\theta^*\|^2$ (shrinkage of the signal).
- Variance $= \sigma^2 \operatorname{tr}(S(S + \lambda I)^{-2})$, bounded when $\lambda > 0$.

Ridge/MAP reduces the variance spike at $d \approx N$.

[7] https://arxiv.org/abs/1612.01474
[8] https://arxiv.org/abs/2107.07511.

5.3 Ridge/MAP: Out-of-Distribution (OOD) Risk

OOD test:

$$x_{\text{test}} = x + \delta, \qquad \delta \sim \mathcal{N}(0, \sigma_\delta^2 I_d).$$

Theorem 5.2 (Exact OOD Risk and Gap, Conditional on $\mathbf{X}$)**.** *Let θ^{MAP} be the MAP estimate. Then*

$$\begin{aligned} R_{OOD}^{MAP}(\mathbf{X}) &:= \mathbb{E}_{x,\delta,\epsilon}[(x^T\theta^* - x_{test}^T\theta^{MAP})^2 \mid \mathbf{X}] \\ &= \|(I-A)\theta^*\|^2 + \sigma_\delta^2\|A\theta^*\|^2 + \sigma^2(1+\sigma_\delta^2)\ \mathrm{tr}\big(S(S+\lambda I)^{-2}\big), \qquad (1) \\ \Delta R^{MAP}(\mathbf{X}) &:= R_{OOD}^{MAP}(\mathbf{X}) - R_{ID}^{MAP}(\mathbf{X}) \\ &= \sigma_\delta^2\|A\theta^*\|^2 + \sigma^2\sigma_\delta^2\ \mathrm{tr}\big(S(S+\lambda I)^{-2}\big). \qquad (2) \end{aligned}$$

Proof. Left as an optional exercise. □

- Signal degradation: $\sigma_\delta^2\|A\theta^*\|^2$.
- Variance amplification: $\sigma^2\sigma_\delta^2\ \mathrm{tr}(S(S+\lambda I)^{-2})$.

💡 Same structure as min-norm, but regularization keeps both terms controlled.

5.4 Full Bayes: In-Distribution Predictive Variance

Posterior covariance:

$$\Sigma_{\text{post}} = \sigma^2(\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}.$$

Predictive variance at a single test input x:

$$\mathrm{PredVar}_{\mathrm{ID}}(x) = \mathrm{Var}(y_{\text{new}} \mid x, D) = \sigma^2 + x^\top\Sigma_{\text{post}}x.$$

Theorem 5.3 (Expected Predictive Variance (ID Test))**.** *Under our setting,*

$$\mathbb{E}_x[\mathrm{PredVar}_{ID}(x) \mid \mathbf{X}] = \sigma^2 + \sigma^2\ \mathrm{tr}\big((\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\big).$$

Proof. Left as an optional exercise. □

💡 This decomposes into **aleatoric** noise (σ^2) plus **epistemic** uncertainty from limited data.

5.5 Full Bayes: OOD Predictive Variance

OOD test input:

$$x_{\text{test}} = x + \delta, \qquad x \sim \mathcal{N}(0, I_d), \qquad \delta \sim \mathcal{N}(0, \sigma_\delta^2 I_d).$$

OOD variance increase:

$$\Delta V_{\text{pred}}(\mathbf{X}) = \mathbb{E}_{x,\delta}[\mathrm{PredVar}_{\mathrm{OOD}}(x+\delta) \mid \mathbf{X}] - \mathbb{E}_x[\mathrm{PredVar}_{\mathrm{ID}}(x) \mid \mathbf{X}].$$

Theorem 5.4 (Exact OOD Predictive Variance Increase)**.** *Under our setting,*

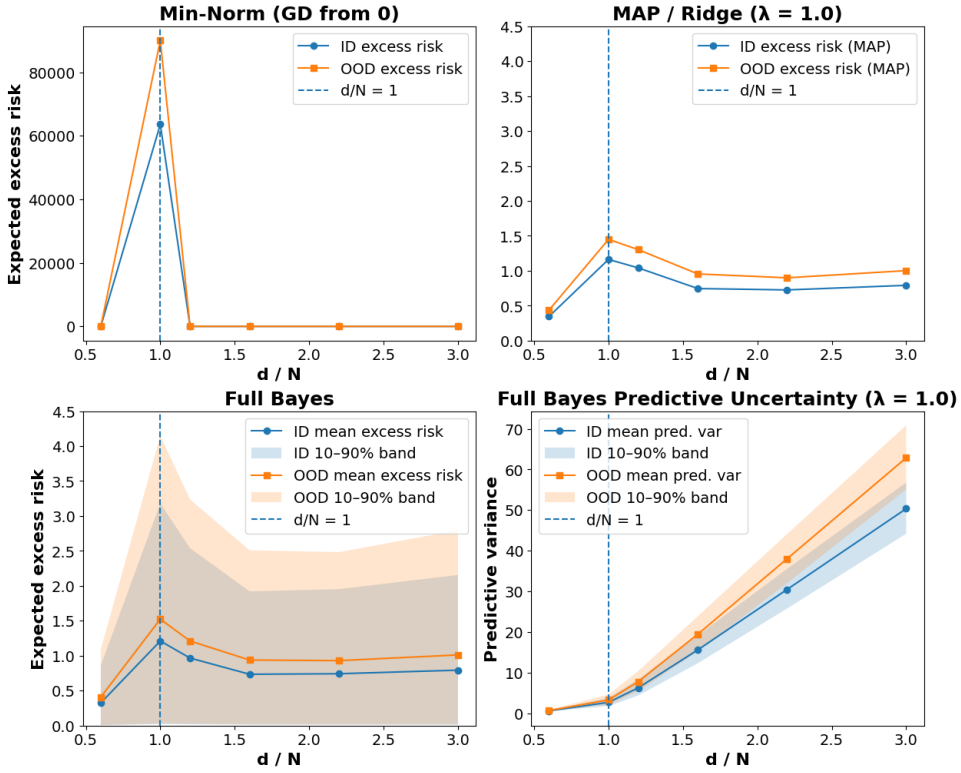
$$\Delta V_{pred}(\mathbf{X}) = \sigma^2 \sigma_\delta^2 \operatorname{tr}\big((\mathbf{X}^\top \mathbf{X} + \lambda I)^{-1}\big).$$

Proof. Left as an optional exercise. □

💡 Full Bayes automatically raises predictive uncertainty under covariate shift — the model "knows when it does not know."

5.6 Empirical Comparison

Setup: $N = 100$, $\sigma = 0.5$, $\sigma_\delta = 0.5$.



5.7 Summary: Three Approaches

So far we have analyzed:

1. **Min-norm (GD from 0)**: no regularization, double descent variance spike.
2. **Ridge/MAP**: adds Gaussian prior, stabilizes variance but only point estimate.
3. **Full Bayes**: posterior predictive, incorporates epistemic uncertainty.

Method	ID Risk	OOD Gap	Uncertainty?
Min-norm	Spike at $d \approx N$	Amplified	None
Ridge/MAP	Bounded	Controlled	Point estimate only
Full Bayes	Bounded	Controlled	Yes

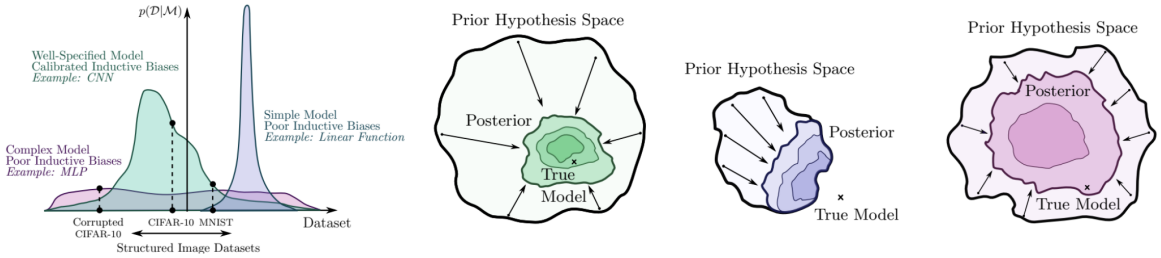
- Regularization ($\lambda > 0$) reduces the double descent spike.
- Full Bayes adds epistemic uncertainty as a reliability signal.

6 Re-Rethinking Generalization?

6.1 Wilson & Izmailov Perspective

Wilson & Izmailov[9] **perspective:** The ability for a system to learn is determined by its support (which solutions are a priori possible) and inductive biases (which solutions are a priori likely).

> Neural networks can fit random labels yet generalize well on real data. This is not paradoxical if we separate **flexibility** (what the model can represent) from **complexity** (what it is biased toward representing).



Bayesian Deep Learning and a Probabilistic Perspective of Generalization
Wilson and Izmailov, 2020

6.2 Support and Inductive Bias

Examples:

- **Linear model:** narrow support, strong but simple inductive bias.
- **CNN:** broad support, inductive bias well-aligned with vision (translation invariance).

Implications:

- Memorization of random labels = broad support without bias.
- Good generalization requires *both* support and inductive bias.
- Bayesian averaging adds another mechanism: smoothing double descent and improving reliability.

💡 This connects back to our Bayes-extended linear–Gaussian analysis.

[9] See `https://arxiv.org/abs/2002.08791`, `https://arxiv.org/pdf/2503.02113`.

7 Exercises

1. Prove the OOD excess-risk theorem.
2. **MAP vs Bayesian (Biased Coin).** Suppose we flip a biased coin with probability λ of landing tails. Let the dataset be $D = \{y_1, \ldots, y_n\}$ where $y_i \in \{0, 1\}$ denotes heads (0) or tails (1).
   (a) Write down the likelihood $p(D \mid \lambda)$ and derive the MLE $\hat{\lambda}_{\text{MLE}}$.
   (b) Suppose we observe $n = 3$ flips: $D = \{1, 1, 0\}$. What is $P(y_4 = 1 \mid D)$ under the MLE estimate? What is it under the Bayesian posterior predictive with prior $\lambda \sim \text{Beta}(a, b)$?
   (c) Analyze the limits:
      – As $a, b \to 0^+$, what happens to the posterior predictive?
      – As $a, b \to \infty$ with $a/(a+b) \to \lambda_0$, what happens?
      – As $n \to \infty$, how do the Bayesian and MLE predictions compare?
   (d) Show that the MAP estimator
   $$\hat{\lambda}_{\text{MAP}} = \arg\max_{\lambda} \log p(\lambda \mid D) = \arg\max_{\lambda} \{\log p(D \mid \lambda) + \log p(\lambda)\}$$
   with a uniform prior $p(\lambda) \propto 1$ coincides with the MLE. Compare this to the Bayesian posterior mean
   $$\mathbb{E}[\lambda \mid D] = \int_0^1 \lambda \, p(\lambda \mid D) \, d\lambda.$$
3. Consider the linear model $\mathbf{y} = \Phi\beta + \boldsymbol{\epsilon}$, where $\Phi \in \mathbb{R}^{n \times m}$. Treat the precision $\tau = 1/\sigma^2$ as unknown with prior $\tau \sim \text{Gamma}(a_0, b_0)$ and conditional prior $\beta \mid \tau \sim \mathcal{N}(0, \tau^{-1}\lambda^{-1}I)$, where λ is fixed.
   (a) Using Bayes' theorem, derive the joint posterior $p(\beta, \tau \mid \mathbf{y})$.
   (b) For a test input x^* with feature vector $\phi^* = \phi(x^*)$, show that the posterior predictive distribution is a Student's t-distribution:
   $$p(y^* \mid x^*, D) = \int p(y^* \mid x^*, \beta, \tau) p(\beta, \tau \mid \mathbf{y}) \, d\beta \, d\tau = t_\nu\big(y^* \mid \mu^*, s^*\big)$$
   where $\nu = 2a_n$. Derive μ^* and s^*.
   (c) Show that as $a_0, b_0 \to \infty$ with $a_0/b_0 \to \tau_0$, the predictive distribution converges to the Gaussian case with $\sigma^2 = 1/\tau_0$.
4. You observe $n = 3$ i.i.d. samples $\mathbf{x} = \{0, 0, 0\}$ from $\mathcal{N}(\mu, 1)$, where μ is unknown. Your prior is
   $$p(\mu) = \tfrac{1}{2}\mathcal{N}(-2, 0.5^2) + \tfrac{1}{2}\mathcal{N}(+2, 0.5^2).$$
   (a) For each component $j \in \{1, 2\}$, use Gaussian conjugacy to compute the updated posterior parameters.
   (b) Compute the posterior mixture weights w_1, w_2 and write the full posterior.

(c) Identify all local maxima of $p(\mu \mid \mathbf{x})$ and determine $\hat{\mu}_{\text{MAP}}$.

(d) Compute the full Bayesian predictive mean $\mathbb{E}[x_4 \mid \mathbf{x}]$ and explain why this differs from using $\hat{\mu}_{\text{MAP}}$.

5. Fit a degree-4 polynomial $f(x) = \sum_{j=0}^{4} a_j x^j$ to $n = 5$ noisy observations. Let $\Phi \in \mathbb{R}^{5\times 5}$ be the design matrix. Assume

$$\mathbf{y} = \Phi\mathbf{a} + \boldsymbol{\epsilon}, \qquad \mathbf{a} \sim \mathcal{N}(0, \alpha^2 I), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I).$$

(a) Show that the MAP estimate equals the ridge regression solution:

$$\hat{\mathbf{a}}_{\text{MAP}} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top \mathbf{y}, \qquad \lambda = \sigma^2/\alpha^2.$$

(b) Derive the full Bayesian posterior $\mathbf{a} \mid \mathbf{y} \sim \mathcal{N}(\mu_n, \Sigma_n)$, and show that for a test point x^* with feature vector ϕ^*,

$$p(y^* \mid x^*, D) = \mathcal{N}\big(\phi^{*\top}\mu_n,\ \sigma^2 + \phi^{*\top}\Sigma_n\phi^*\big).$$

(c) Explain why the epistemic uncertainty term $\phi^{*\top}\Sigma_n\phi^*$ grows as x^* moves away from the training data support.

A Appendix 1: Model Uncertainty and Calibration

A.1 Trustworthy ML in the Real World

Real-world deployment scenarios:

- Medical diagnosis: "How confident is the model?"
- Autonomous driving: "Should I brake or continue?"
- Financial trading: "What is the risk of this prediction?"
- Scientific discovery: "How much should we trust this result?"

Central questions:

1. Does the model generalize to new distributions? (OOD)
2. Does it know what it does not know? (Uncertainty)
3. Are its confidence scores honest? (Calibration)
4. Can it handle perturbations? (Robustness)

⚠ **Neural networks**[10] are often overconfident, especially on incorrect predictions.

A.2 Model Uncertainty and Calibration

Two sources of predictive uncertainty.

Condition on the observed dataset $D = (\mathbf{X}, \mathbf{y})$. For a new input $\mathbf{x}_*$ and model parameter θ,

$$\mathrm{Var}[Y_* \mid \mathbf{x}_*, D] = \underbrace{\mathbb{E}_{\theta|D}\big[\mathrm{Var}\left(Y_* \mid \mathbf{x}_*, \theta\right)\big]}_{\textbf{Aleatoric}} + \underbrace{\mathrm{Var}_{\theta|D}\big(\mathbb{E}\big[Y_* \mid \mathbf{x}_*, \theta\big]\big)}_{\textbf{Epistemic}}.$$

- **Aleatoric** uncertainty:
  - Irreducible data noise.
  - Cannot be reduced by more data.
- **Epistemic** uncertainty:
  - Parameter/model uncertainty due to limited data or misspecification.
  - Can shrink with more data or better inductive bias.

A.3 Model Calibration: The Problem

Calibration[11]**:** predicted probabilities should match actual frequencies.

Perfect calibration: if a model predicts 80% confidence, it should be correct 80% of the time.

Modern deep networks are poorly calibrated[12]**:**

- Often overconfident.
- Larger networks tend to be more overconfident.
- Temperature scaling is commonly used as a post-processing fix.

[10] See `https://www.cs.ox.ac.uk/people/yarin.gal/website/thesis/thesis.pdf`.
[11] See `https://arxiv.org/pdf/2501.19047` for an intro.
[12] See `https://arxiv.org/abs/1706.04599`.

Consequences:

- Poor decision making in critical applications.
- Inability to reject uncertain predictions.
- False sense of model reliability.

A.4 Expected Calibration Error (ECE)

Definition A.1 (Expected Calibration Error)**.**

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$$

where:

M bins of predictions,

- B_m = set of samples with confidence in bin m,
- $\text{acc}(B_m)$ = accuracy of predictions in bin m,
- $\text{conf}(B_m)$ = average confidence in bin m,
- n = total number of samples.

Lower ECE = better calibration.

⚠ Limitations: binning dependency, sample size effects.

A.5 Calibration Methods: Post-hoc Fixes

Adjust predicted probabilities so that confidence matches accuracy.

Post-hoc calibration: learn a mapping from raw model outputs (logits) to calibrated probabilities using a held-out validation set.

Common methods:

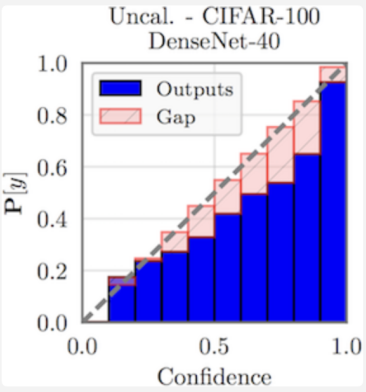
- **Temperature scaling:**
$$p_T(y|x) = \frac{\exp(z_y/T)}{\sum_k \exp(z_k/T)}.$$
- Others: histogram binning, Platt scaling, isotonic regression.

Limitations:

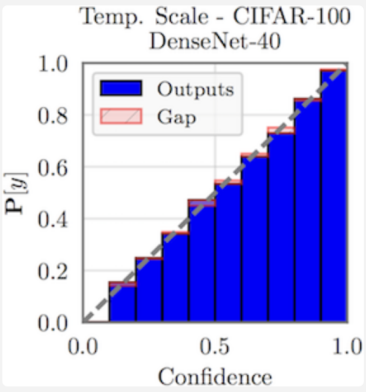
- Only adjust outputs *after training*.
- Do not fix the root cause of overconfidence.
- Accuracy is unchanged, calibration is improved.

A.6 Reliability Diagram

Since confidence should reflect accuracy, we would like the plot in the reliability diagram to be the identity function.



An uncalibrated neural network, before temperature scaling. The reliability diagram indicates miscalibration.



Neural network after temperature scaling. The reliability diagram indicates a well-calibrated network.

ECE and reliability diagrams give practical diagnostics, but they are ad hoc and depend on binning choices. A more principled way to evaluate calibrated probabilities is through **proper scoring rules**.

A.7 Proper Scoring Rules

A scoring rule $S(q, y)$ evaluates the quality of a probabilistic prediction q when the true outcome is y.

Properness: A scoring rule is **proper** if the best strategy is to report the true distribution p:

$$\mathbb{E}_{Y\sim p}[S(p, Y)] \leq \mathbb{E}_{Y\sim p}[S(q, Y)] \quad \forall q.$$

It is **strictly proper** if equality holds only when $q = p$.

Examples:

- **Logarithmic score / NLL:**
$$S(q, y) = -\log q(y).$$
- **Brier score:**
$$S(q, y) = \sum_k (q_k - \mathbf{1}\{y = k\})^2.$$
- **CRPS:**
$$S(q, y) = \int_{-\infty}^{\infty} \left(F_q(z) - \mathbf{1}\{y \leq z\}\right)^2 dz.$$

💡 Proper scoring rules reward *honest, calibrated probabilities.*

B Appendix 2: Model Robustness

B.1 Model Robustness

Adversarial example[13]**:** for input x, find δ with $\|\delta\| \ll 1$ such that $f(x) \neq f(x + \delta)$.

[13] Picture from `https://arxiv.org/abs/1412.6572`. See also `https://arxiv.org/pdf/1312.6199`.

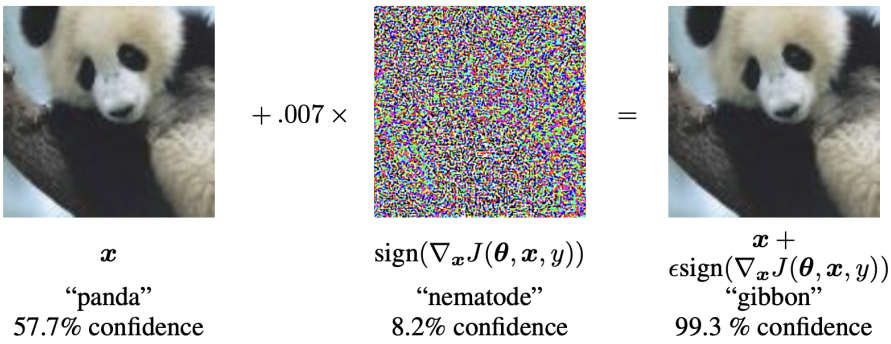


Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

- Despite high i.i.d. accuracy, decision boundaries are fragile.
- Neural networks have "blind spots": tiny perturbations cause confident misclassifications.

⚠ Good average performance $\neq$ reliable predictions.

B.2 The Fragility Problem

Adversarial fragility is largely *orthogonal*[14] to the overparameterization puzzle.

Fragility exists across model types:

- Linear classifiers are adversarially vulnerable.
- Small neural networks can be fragile.
- Even well-regularized models suffer from adversarial examples.

Fundamental causes:

- High-dimensional input spaces enable small perturbations.
- Decision boundaries concentrate near the data manifold.
- Human perception vs. ℓ_p-norm misalignment.

Connection to reliability:

- Good average performance $\neq$ robust performance.
- Models can be simultaneously accurate and fragile.
- Robustness requires explicit consideration, not just scale.

B.3 Robustness to Natural Corruptions

See `https://arxiv.org/abs/1903.12261`.

Beyond adversarial: models degrade under natural corruptions:

- Noise, blur, weather effects.
- JPEG compression artifacts.

[14] There are claims that simplicity bias could lead to such vulnerability; see `https://arxiv.org/abs/2006.07710`.

- Lighting changes.

Benchmarks:

- CIFAR-10-C, ImageNet-C, etc.
- Mean Corruption Error (mCE) relative to baseline.

Bridge to reliability: robustness is not only worst-case (adversarial) but also average-case under realistic distribution shifts.

B.4 Data Augmentation for Robustness

Key idea: robustness is not only about *defenses*, but also about *exposing models to richer data distributions.*

- **Basic augmentation:** flips, crops, noise injection.
- **Advanced augmentation:** Mixup, Noisy Feature Mixup, AugMix.
- **Statistical view:** augmentation $\approx$ sampling from an invariance prior.

FIGURE 10.9. *Data augmentation. The original image (leftmost) is distorted in natural ways to produce different images with the same class label. These distortions do not fool humans, and act as a form of regularization when fitting the CNN.*

💡 Data augmentation can improve both *generalization* and *robustness.*

B.5 Beyond Classical Augmentation: Mixup and Variants

Classical augmentation preserves the original label.

Modern methods interpolate between data points:

- **Mixup (Zhang et al., 2017):**

$$\tilde{x} = \lambda x_1 + (1-\lambda)x_2, \qquad \tilde{y} = \lambda y_1 + (1-\lambda)y_2,$$

  where $\lambda \sim \text{Beta}(\alpha, \alpha)$.

- **CutMix (Yun et al., 2019):** replace a random patch of x_1 with x_2, assign mixed label.

These encourage smooth transitions between classes and improve robustness.

💡 *Bayesian view: Mixup can be seen as marginalization over input neighborhoods.*

C From Linear Models to Large Language Models (LLMs)?

The same or analogous phenomena we saw in linear models reappear in LLMs.

Overparameterization and generalization:

- Billions of parameters ⇒ extreme overparameterization.
- Like min-norm regression, they can memorize training data.
- Yet architectures (transformers) provide strong inductive bias, enabling generalization.

Reliability challenges:

- **Hallucinations:** confident but wrong outputs.
- **Miscalibration:** probabilities do not reflect true correctness.
- **Prompt sensitivity:** small input perturbations ⇒ large changes.
- **Distribution shift:** performance degrades on unfamiliar domains, just like OOD test inputs.

C.1 The Forbidden Knowledge Map of LLMs

⚠ The intuition starts with linear models, but LLMs add many layers of complexity.

Foundations

- Text → tokens → embeddings
- Positional information
- Self-attention with Q, K, V
- Multihead attention
- Transformer block = attention + MLP + norm

Decoding & Memory

- Sampling: temperature, top-k, top-p
- KV cache
- Long context methods
- Mixture of Experts
- GQA
- LayerNorm, RMSNorm
- GELU, SiLU, ReLU

Training & Deployment

- Objectives: causal LM, masked LM
- Fine-tuning, instruction tuning
- Preference optimization: RLHF, DPO
- Scaling laws
- Quantization
- Inference systems: vLLM, TGI, TensorRT-LLM
- Synthetic data generation