OCTA Research – 365 Days of Learning, Theorizing, and Building AGI

Singular Scaling Laws for Transformers: RLCT, Exponents, and Capability Transitions

Week 1 · Day 5 · January 5, 2026

OCTA Research Internal Theory Program

Version 1.1 – Living Document Series (Expanded)

OCTA Research 365 Program Note (Week 1, Day 5). Day 1 separated macro "Singularity" narratives from micro singularities in architectures. Day 2 built the stratified geometry of attention (tie manifolds, block cells, parameter varieties). Day 3 introduced Singular Learning Theory (SLT) and the Real Log Canonical Threshold (RLCT) λ as effective dimension. Day 4 translated SLT into Fisher geometry and RLCT *probes* for real Transformers. Day 5 asks: *How do these singular-geometry quantities control scaling laws and capability transitions?* We synthesize classical neural scaling, the RLCT picture, and OCTA's Fisher instrumentation into a unified *singular scaling surface* that can be measured, fitted, and used as a control interface.

Abstract

Empirical scaling laws show that Transformer performance often follows approximate power laws in model size, dataset size, and compute. Singular Learning Theory (SLT) and Fisher geometry suggest that these exponents encode the effective dimension and singular structure of the model. Day 5 develops a singular scaling framework for Transformers:

- we recap classical scaling laws (loss vs. dataset, parameters, and compute);
- we insert RLCT and Fisher-based effective dimension into these laws, clarifying when exponents can be interpreted as geometry-normalized;
- we define *singular scaling surfaces* with multiple regimes and crossovers, and introduce OCTA scaling phases (Phase 0–3);
- we connect sharp changes in exponents to capability transitions and attention strata;
- we propose OCTA protocols for multi-axis scaling sweeps, piecewise power-law fitting, geometry-conditioned regime analysis, and capability transition mapping;
- we sketch an OCTA *Scaling Surface Dashboard* and control logic for safety windows and resource-optimal scaling paths;
- we anchor intuition with a two-regime toy model that exhibits a genuine change in effective dimension.

The goal is to treat scaling exponents, RLCT, Fisher spectra, and capability loci as a unified diagnostic surface on which OCTA can steer architectures, training policies, and safety margins.

Contents

List of Figures

1 Overview: Scaling Laws Meet Singular Geometry

Scaling laws for language models assert that, over wide ranges, test loss L or negative log-likelihood obeys approximate power laws in:

- dataset size N (number of tokens or examples),
- model size P (parameter count),
- and compute C (e.g. FLOPs).

A prototypical form is

$$L(N) \approx L_\infty + AN^{-\alpha}, \qquad L(P) \approx L_\infty + BP^{-\beta}, \tag{1}$$

for exponents $\alpha, \beta > 0$ in a given regime.

Day 3 showed that in singular models the generalization gap G_N behaves asymptotically as

$$\mathbb{E}[G_N] = \mathbb{E}\left[D_{\mathrm{KL}}\big(q \,\|\, p_{\hat{\theta}_N}\big)\right] \approx \frac{\lambda}{N}, \tag{2}$$

with RLCT λ replacing the usual $d/2$. Day 4 provided tools for estimating Fisher spectra and RLCT proxies.

Day 5 integrates these layers:

- **Theory:** scaling exponents as functions of RLCT and effective dimension.
- **Geometry:** multiple scaling regimes as different geometry phases in parameter space.
- **Empirics:** protocols to fit piecewise power-law surfaces and track geometry parameters across regimes.

- **Control:** using scaling surfaces and capability loci as steering tools in OCTA.

We move systematically from classical single-exponent laws to a *multi-regime singular scaling surface*.

2 Classical Neural Scaling Laws (Recap)

We briefly recall the main ingredients of classical scaling laws to fix notation.

2.1 Loss vs. dataset size

For a fixed model architecture and training protocol, as dataset size N grows, test loss $L(N)$ often follows

$$L(N) \approx L_\infty(M) + A(M)N^{-\alpha(M)}, \tag{3}$$

where M denotes model size or architecture, and:

- $L_\infty(M)$ is an irreducible loss floor (due to model misspecification or task noise);
- $A(M)$ is a prefactor;
- $\alpha(M)$ is the dataset-size exponent, often roughly stable in a given regime.

On log–log axes, $(\log N, \log(L(N) - L_\infty))$ is approximately linear with slope $-\alpha(M)$.

2.2 Loss vs. model size

For fixed dataset size N and training protocol, loss often scales as

$$L(P) \approx L_\infty(N) + B(N)P^{-\beta(N)}, \tag{4}$$

with $\beta(N)$ capturing how rapidly adding parameters improves performance.

2.3 Compute-optimal tradeoffs

Let C denote total training compute. Under simple assumptions (e.g. $C \approx kNP$ for some constant k), one can derive compute-optimal paths in (N, P) space that minimize loss for a given compute budget C_0.

The classical view treats (α, β) as empirical knobs; Day 5 asks how they relate to RLCT λ and singular geometry.

3 Singular Learning Theory View of Scaling

Recall from Day 3: for a singular statistical model with RLCT λ and multiplicity m, the Bayesian generalization behavior in the large-sample limit satisfies

$$\mathbb{E}[G_N] = \mathbb{E}\big[D_{\mathrm{KL}}(q \,\|\, p_{\hat{\theta}_N})\big] \approx \frac{\lambda}{N} + \frac{m-1}{N \log N} + o\left(\frac{1}{N}\right), \tag{5}$$

under suitable regularity conditions and priors.

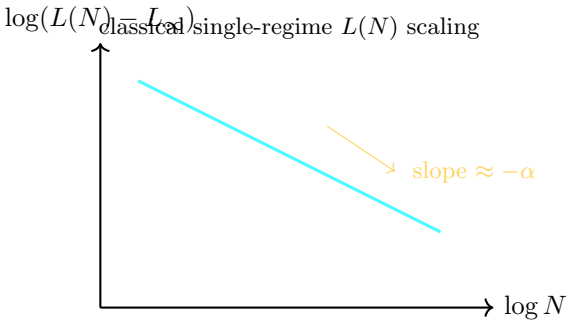

Figure 1: Classical neural scaling: log–log plots of $(L(N) - L_\infty)$ vs. N often show approximately straight-line behavior over wide ranges, with slope $-\alpha$.

3.1 Regular vs. singular scaling coefficients

Regular models. If the model is regular with parameter dimension d and Fisher information positive-definite at the minimizer, then:

- RLCT $\lambda = d/2$,
- multiplicity $m = 1$,
- the leading term is $d/(2N)$.

Singular models. In singular models:

- λ can be much smaller than $d/2$;
- m can be greater than 1;
- the asymptotic generalization curve is still $\sim \lambda/N$, but with a rich subleading structure.

Conceptually:

- redundancy, symmetries, and strata reduce the effective dimension;
- the algebraic structure of the parameter set near the minimizer determines λ;
- scaling is controlled by geometry, not raw parameter count.

3.2 Effective exponents in non-asymptotic regimes

In practice, experiments often operate in non-asymptotic regimes where:

- N is large but not infinite;
- optimization noise, regularization, and architectural constraints matter;
- $L(N)$ is closer to $L_\infty + AN^{-\alpha}$ with $\alpha \neq 1$.

We can view α as an *effective exponent* that blends:

- the asymptotic $1/N$ SLT behavior;

- non-asymptotic corrections (e.g. additional power-law terms);
- structural changes in RLCT as we move across regimes.

Definition 3.1 (RLCT-informed exponent family)**.** Given a model family and a regime in which

$$L(N) \approx L_\infty + AN^{-\alpha},$$

define:

- an RLCT proxy λ_{eff} via Day 4 methods (learning curves and Fisher geometry);
- a normalized exponent $\alpha^\star$ via
$$\alpha^\star := \alpha \cdot \frac{NG_N}{\lambda_{\text{eff}}}, \tag{6}$$
where G_N is an empirical generalization gap at some reference N in the regime.

We say the regime is *SLT-compatible* if $\alpha^\star \approx 1$ and relatively stable across nearby architectures.

Intuitively, once we factor out λ_{eff}, the remaining scaling structure behaves as if the asymptotic $1/N$ law were in effect.

OCTA Principle 3.2 (OCTA Scaling Principle I – Geometry-normalized scaling). If two architectures have Fisher/RLCT proxies $\lambda_{\text{eff}}^{(1)}$ and $\lambda_{\text{eff}}^{(2)}$, then their dataset-size scaling should be compared only after normalizing by λ_{eff}:

$$\tilde{L}_i(N) := \frac{L_i(N) - L_{\infty,i}}{\lambda_{\text{eff}}^{(i)}} \approx \tilde{A}N^{-\alpha^\star},$$

where $\tilde{A}$ and $\alpha^\star$ are shared, geometry-normalized quantities.

In OCTA, this says: treat λ_{eff} (and related Fisher metrics) as "scale factors" that renormalize scaling exponents across architectures.

4 Singular Scaling Surfaces and OCTA Phases

Real models rarely live in a single clean asymptotic regime; instead, they exhibit *multiple scaling regimes* with crossovers.

4.1 Singular scaling surface: formalization

Let N denote dataset size, P model parameters, and C compute. We define:

Definition 4.1 (Singular scaling surface)**.** Let $L(N, P)$ be test loss as a function of (N, P) under a fixed training protocol and data distribution. Define the *singular scaling surface* as

$$\mathcal{S} := \{(N, P, L(N, P)) \in \mathbb{R}^3_+\}.$$

Equip each point (N, P) with:

- local exponents $(\alpha(N, P), \beta(N, P))$ extracted from small neighborhoods:
$$L(N', P) \approx L(N, P) + A(N, P)\big((N')^{-\alpha(N,P)} - N^{-\alpha(N,P)}\big),$$
and analogously for variation in P;

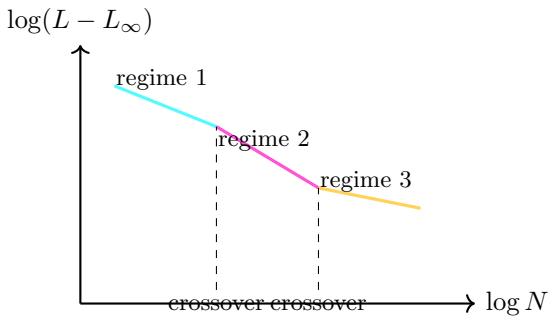

Figure 2: Multi-regime scaling: log–log loss vs. dataset size with distinct linear segments. In the singular view, each regime has its own approximate RLCT-like coefficient λ_r and Fisher geometry.

- local RLCT proxy $\lambda_{\text{eff}}(N, P)$ (Day 4);
- local Fisher geometry descriptors (spectrum entropy H_{spec}, top eigenvalues, headwise traces).

We say $\mathcal{S}$ is *stratified* if it decomposes into regions

$$\mathcal{S} = \bigcup_r \mathcal{S}_r$$

such that each region r has approximately constant $(\alpha_r, \beta_r, \lambda_r)$ and stable geometry descriptors, while different regions have distinct tuples.

4.2 OCTA scaling phases (Phase 0–3)

For operational use, we introduce a coarse taxonomy:

Definition 4.2 (OCTA scaling phases)**.** For a given architecture family, we define four canonical phases as N and P increase:

- **Phase 0 (underparameterized / undertrained):**
  - small P and/or small N;
  - high loss, weak approximation;
  - Fisher spectra low-energy and noisy; RLCT proxies unstable.
- **Phase 1 (emergent power-law regime):**
  - intermediate N and P;
  - clear power-law region in $L(N)$ and $L(P)$;
  - Fisher spectra exhibit stable shape; RLCT proxies well-defined.
- **Phase 2 (capability-rich scaling):**
  - larger N and P, with several capabilities active;
  - multiple scaling segments with different $\alpha^{(r)}, \beta^{(r)}$;
  - Fisher spectra and RLCT proxies shift as new strata become active.

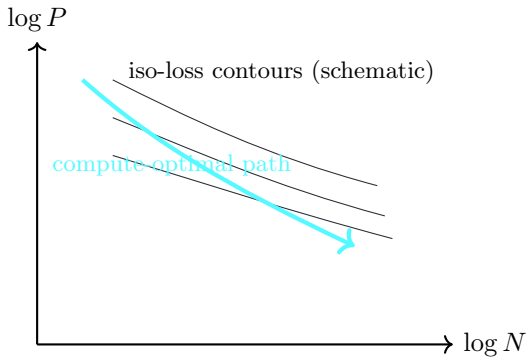

Figure 3: Conceptual $(\log N, \log P)$ scaling surface: iso-loss contours (grey) and an example compute-optimal path (cyan). Different regions along this path can belong to different OCTA phases.

- **Phase 3 (saturation / task-limited):**
  - $L(N, P)$ approaches a floor given task noise or model class limitations;
  - scaling gains diminish; effective exponents shrink;
  - geometry may still change, but performance gains are small.

OCTA Principle 4.3 (OCTA Scaling Principle II – Phase-aware interpretation). Scaling laws must be interpreted phase-by-phase:

- exponents (α, β), RLCT proxies, and Fisher spectra are meaningful only within relatively stable phases;
- comparisons between architectures and training recipes should be conditioned on being in the same phase;
- OCTA experiments should explicitly label which phase each (N, P) configuration occupies.

4.3 Visualizing the 2D scaling surface

It is useful to visualize the full $(\log N, \log P)$ plane with loss-level contours.

5 Capability Transitions as Singular Events

Capabilities (e.g. in-context learning, multi-step reasoning, tool use) often appear abruptly as N, P, or C cross certain thresholds. In the singular picture:

- capabilities are associated with new strata in function space becoming accessible;
- the model transitions from one region of the scaling surface to another with different RLCT and Fisher geometry;
- performance metrics on specific tasks show sharp changes in scaling exponents or intercepts.

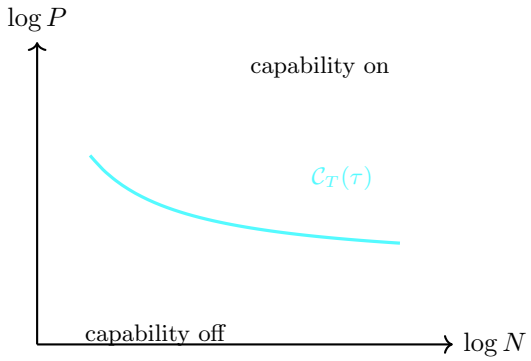

Figure 4: Capability transition locus: for a given task T and threshold τ, a curve in $(\log N, \log P)$ space separates regions where the capability is typically off vs. on.

5.1 Capability indicator functions and loci

Let T index tasks or capability metrics (e.g. pass@k on code tasks, accuracy on reasoning tasks). Define capability curves:

$$\mathrm{Cap}_T(N, P) := \text{task-specific metric for task } T, \tag{7}$$

with values typically in $[0, 1]$.

Definition 5.1 (Capability transition locus)**.** For a task T and threshold τ (e.g. 0.5 accuracy, or 0.1 pass@k), define the capability transition locus

$$\mathcal{C}_T(\tau) = \{(N, P) : \mathrm{Cap}_T(N, P) = \tau\}.$$

We think of $\mathcal{C}_T(\tau)$ as a curve on the scaling plane where capabilities "turn on."

5.2 Singular interpretation of capability loci

In the SLT view:

- $\mathcal{C}_T(\tau)$ corresponds to crossing from one effective model class to another, where the model's function space includes qualitatively different solutions;
- RLCT proxies and Fisher spectra typically show changes across $\mathcal{C}_T(\tau)$:
  - new directions become active in Fisher spectra;
  - attention-strata distributions shift (e.g. increased use of long-range attention patterns identified in Day 2);
  - headwise Fisher blocks change from "dormant" to "active" geometry.

OCTA Principle 5.2 (OCTA Scaling Principle III – Capabilities as geometry bifurcations). Capability transitions should be interpreted as *bifurcations in the singular geometry*:

- the function manifold accessible to the model undergoes a topological or stratification change;
- this is detectable via changes in Fisher rank, spectrum anisotropy, headwise traces, and strata occupancy;

- scaling exponents (α, β) can change across these bifurcations.

OCTA scaling experiments should therefore pair task metrics with geometry probes at and around $\mathcal{C}_T(\tau)$.

6 OCTA Protocols: Singular Scaling Experiments

We now specify concrete OCTA protocols for measuring scaling surfaces, exponents, and capability loci, while logging singular geometry.

6.1 Protocol D5.1: Multi-axis scaling sweep

Experimental Protocol 6.1 (D5.1: Multi-axis (N, P) scaling sweep).

(a) Choose:

- a set of model sizes $\{P_i\}_{i=1}^{M}$ (e.g. varying depth, width, head count);
- a set of dataset sizes $\{N_j\}_{j=1}^{K}$ (e.g. geometric progression);
- a training recipe held as constant as possible across (N_j, P_i).

(b) For each pair (N_j, P_i):

- train a model to near convergence using dataset size N_j and architecture with P_i;
- record test losses L_{ij} and task-specific capabilities $\mathrm{Cap}_T(N_j, P_i)$ for a suite of tasks T;
- run Day 4 Fisher probes at one or more checkpoints (e.g. late training);
- tag each run with an OCTA phase (Phase 0–3) based on loss, capabilities, and geometry indicators.

(c) Assemble:

- a matrix L of losses;
- a tensor of capability scores;
- a dataset of Fisher spectra and RLCT proxies at each (N_j, P_i);
- a phase label for each point.

6.2 Protocol D5.2: Piecewise power-law fitting with breakpoints

Experimental Protocol 6.2 (D5.2: Piecewise power-law fitting).

(a) For each fixed P_i, fit L_{ij} vs. N_j on log–log axes using:

- (i) single power-law model $L(N) \approx L_{\infty,i} + A_i N^{-\alpha_i}$;
- (ii) two or three segment piecewise-linear models with breakpoints:

$$\log(L(N) - L_{\infty,i}) \approx \begin{cases} a_1 + b_1 \log N & N \le N_{\text{break}}^{(1)} \\ a_2 + b_2 \log N & N_{\text{break}}^{(1)} < N \le N_{\text{break}}^{(2)} \\ \dots \end{cases}$$

(b) Use goodness-of-fit and model selection (e.g. AIC/BIC, cross-validation) to identify:

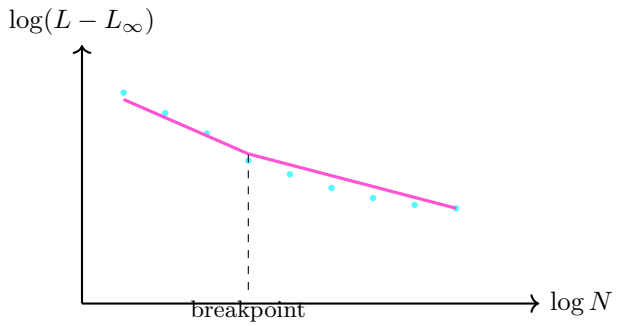

Figure 5: Illustration of piecewise power-law fitting (D5.2): data points (cyan) are fit by two segments (magenta) with a breakpoint where the scaling exponent changes.

- number of regimes per P_i;
- exponents $\alpha_i^{(r)} = -b_r$ and breakpoints $N_i^{(r)}$.

(c) For each regime r, average the exponents across P_i (or across regimes that align) to obtain regime-level exponents $\alpha^{(r)}$.

(d) Jointly inspect Fisher and RLCT proxies from Day 4 at breakpoints to see if:

- Fisher spectra change shape (e.g. sudden shift in top eigenvalues);
- effective dimension $d_{\text{eff}}(\alpha)$ or λ_{eff} changes;
- phase labels (Phase 1 $\to$ Phase 2, etc.) change.

6.3 Protocol D5.3: Geometry-conditioned scaling analysis

Experimental Protocol 6.3 (D5.3: Geometry-conditioned scaling).

(a) For each scaling regime identified by D5.2, compute:

- average Fisher spectra and entropy H_{spec};
- average effective dimension $d_{\text{eff}}(\alpha)$;
- RLCT proxies λ_{eff} (from learning curves as in Day 4);
- distribution of OCTA phases (fraction of runs in Phase 1, Phase 2, etc.).

(b) For each breakpoint:

- inspect changes in these geometry metrics;
- label breakpoints as:
  - *soft geometry change* (small shifts in spectra);
  - *hard geometry change* (large shifts in rank, anisotropy, or RLCT).

(c) Build a table of regimes:

$$\mathcal{R}_r = \left(\alpha^{(r)}, \beta^{(r)}, \lambda_{\text{eff}}^{(r)}, d_{\text{eff}}^{(r)}, H_{\text{spec}}^{(r)}, \text{Phase}_r\right),$$

and interpret each as a geometry phase with associated scaling behavior.

6.4 Protocol D5.4: Capability scaling and transition maps

Experimental Protocol 6.4 (D5.4: Capability transition mapping).

(a) For each task T in a capability suite:

- record $\mathrm{Cap}_T(N_j, P_i)$ across the scaling grid;
- identify thresholds τ of interest (e.g. 0.5 accuracy, 0.1 pass@k).

(b) Approximate the transition locus $\mathcal{C}_T(\tau)$ by interpolation in $(\log N, \log P)$ space.

(c) At points near $\mathcal{C}_T(\tau)$, inspect:

- whether scaling exponents (α, β) change;
- how Fisher spectra and RLCT proxies behave;
- how attention-strata occupancy shifts (Day 2);
- whether OCTA phase labels change (e.g. Phase 1 $\to$ Phase 2).

(d) Classify transitions as:

- *gradual*: capability improves continuously with no sharp geometric changes;
- *singular*: capability improves with an associated hard geometry change and regime switch.

6.5 Protocol D5.5: Scaling surface dashboard

Experimental Protocol 6.5 (D5.5: OCTA Scaling Surface Dashboard).

(a) Build a dashboard that, for a given architecture family, shows:

- heatmaps of $L(N, P)$ and $\mathrm{Cap}_T(N, P)$;
- overlaid capability locus curves $\mathcal{C}_T(\tau)$;
- region-level exponents $(\alpha^{(r)}, \beta^{(r)})$ and breakpoints;
- geometry metrics (Fisher, RLCT proxies) and OCTA phase labels per region.

(b) Allow interactive selection of:

- slices at fixed N or P;
- zoom into neighborhoods near capability transitions;
- overlays of Fisher spectra summaries (e.g. top eigenvalues, entropy);
- overlays of safety windows (Section 9).

7 Implementation Skeletons for Scaling Sweeps

We sketch high-level pseudo-code for running scaling sweeps and logging geometry. These are conceptual; real implementations will depend on the underlying framework.

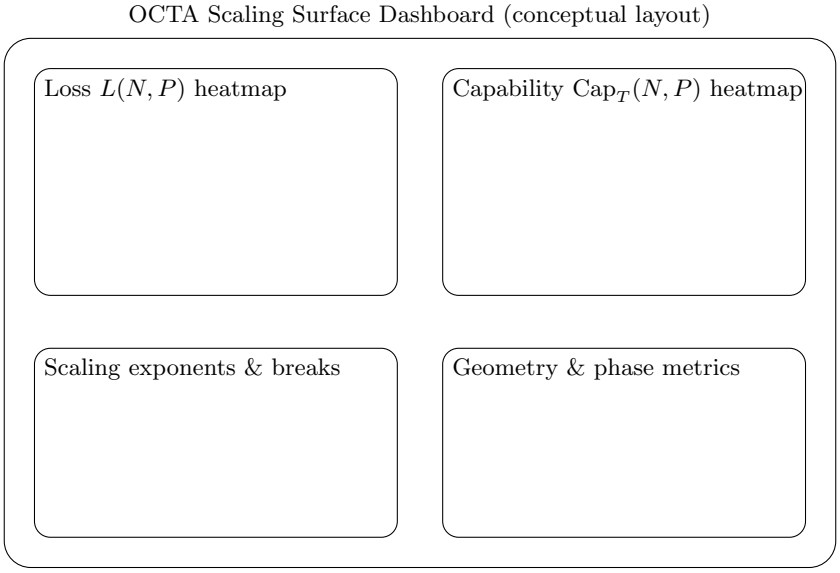

Figure 6: Conceptual OCTA Scaling Surface Dashboard: top row shows loss and capability heatmaps over (N, P); bottom row shows fitted exponents with breakpoints and geometry/phase metrics per regime.

7.1 Scaling grid runner

```
def run_scaling_grid(N_list, arch_list, train_recipe, tasks,
                     fisher_config, phase_criteria):
    results = []
    for P_cfg in arch_list:
        for N in N_list:
            run_id = f"N={N}_arch={P_cfg.name}"
            # 1. Build model
            model = build_model(P_cfg)
            # 2. Prepare dataset subset/loader of size N
            train_loader, val_loader, test_loader = make_dataloaders(N)
            # 3. Train
            model, ckpt, train_log = train_model(
                model, train_loader, val_loader,
                train_recipe, run_id=run_id
            )
            # 4. Evaluate loss
            loss_metrics = eval_loss(model, test_loader)
            # 5. Evaluate capabilities
            cap_metrics = {}
            for T in tasks:
                cap_metrics[T.name] = T.evaluate(model, test_loader)
            # 6. Run Fisher / RLCT probes near end of training
            geom_metrics = run_fisher_probes(
                model, fisher_config, test_loader
            )
```

```
            # 7. Determine OCTA phase (0-3)
            phase = classify_phase(loss_metrics, cap_metrics,
                                   geom_metrics, phase_criteria)
            # 8. Store
            results.append({
                "N": N,
                "arch": P_cfg,
                "loss": loss_metrics,
                "capabilities": cap_metrics,
                "geometry": geom_metrics,
                "phase": phase,
                "train_log": train_log,
                "checkpoint": ckpt,
            })
    return results
```

7.2 Piecewise scaling fit and regime extraction

```
def fit_piecewise_scaling(results, fixed_arch=None, fixed_N=None):
    # Select slices along N or P
    if fixed_arch is not None:
        slice_data = [(r["N"], r["loss"]["test_loss"])
                      for r in results if r["arch"] == fixed_arch]
        x = np.log([N for (N, L) in slice_data])
        base = min(L for (_, L) in slice_data) - 1e-8
        y = np.log([max(L - base, 1e-8) for (_, L) in slice_data])
    elif fixed_N is not None:
        slice_data = [(r["arch"].size, r["loss"]["test_loss"])
                      for r in results if r["N"] == fixed_N]
        x = np.log([P for (P, L) in slice_data])
        base = min(L for (_, L) in slice_data) - 1e-8
        y = np.log([max(L - base, 1e-8) for (_, L) in slice_data])
    else:
        raise ValueError("Specify fixed_arch or fixed_N")

    # Fit 1- and 2-segment models (details omitted)
    single_fit  = fit_single_line(x, y)
    multi_fit   = fit_two_segment_line(x, y)
    best_model  = select_best_model(single_fit, multi_fit)
    return best_model  # contains exponents, breakpoints, fit quality
```

7.3 Geometry-conditioned regime summary

```
def summarize_geometry_by_regime(results, regimes):
    # regimes: list of regime definitions with N/P ranges and phase info
    regime_geom = []
    for reg in regimes:
        reg_points = []
```

```
        for r in results:
            if in_regime(r["N"], r["arch"].size, r["phase"], reg):
                reg_points.append(r["geometry"])
        geom_summary = aggregate_geometry(reg_points)
        regime_geom.append({
            "regime": reg,
            "geometry": geom_summary,
        })
    return regime_geom
```

These sketches capture the core loops for D5.1–D5.3: sweep the scaling grid, fit piecewise exponents, and aggregate geometry metrics per regime, with explicit phase conditioning.

8 Toy Model: Two-Regime Effective Dimension Change

To anchor the singular scaling picture, we present a toy model where the effective dimension (and thus RLCT) changes genuinely between regimes.

8.1 Model definition

Consider a family of models with two blocks:

- a "core" block with parameters $\theta_c \in \mathbb{R}^{d_c}$;
- a "head" block with parameters $\theta_h \in \mathbb{R}^{d_h}$.

Assume:

- the data distribution q can be decomposed as $q = q_{\text{core}} \otimes q_{\text{head}}$;
- the core block is needed for basic performance;
- the head block is required for a higher-level capability that only becomes learnable once the core block is sufficiently trained.

For the sake of the toy model, suppose:

- for small N, the head block is effectively inactive (e.g. its gradients are dominated by noise or regularization);
- for larger N, the head block becomes active and starts to contribute to the likelihood.

8.2 Phase 1: core-only effective dimension

When $N \in [N_{\min}, N_{\text{switch}})$, assume:

- the core block behaves like a regular model with effective dimension d_c;
- the head block is effectively frozen or unused;
- the SLT RLCT is approximately $\lambda_1 \approx d_c/2$.

In this regime, the generalization gap scales as

$$\mathbb{E}[G_N] \approx \frac{\lambda_1}{N} = \frac{d_c/2}{N}.$$

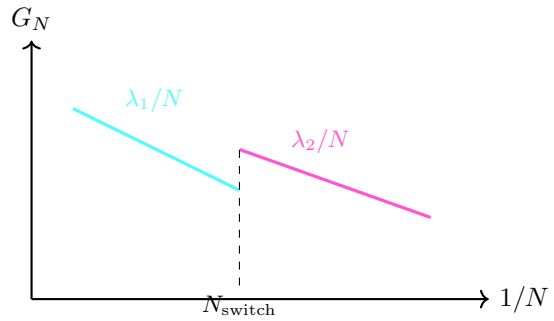

Figure 7: Toy two-regime RLCT change: Phase 1: effective dimension λ_1 (core only); Phase 2: effective dimension $\lambda_2 > \lambda_1$ (core + head). Both regimes have $G_N \sim 1/N$, but prefactors differ.

8.3 Phase 2: core + head effective dimension

When $N \geq N_{\text{switch}}$, the head block becomes active. Assume that, after reparametrization, the joint model behaves like a singular model with effective RLCT λ_2, satisfying

$$\lambda_1 < \lambda_2 \leq \frac{d_c + d_h}{2}.$$

Then, in the second regime:

$$\mathbb{E}[G_N] \approx \frac{\lambda_2}{N},$$

and the slope in G_N vs $1/N$ steepens.

8.4 Connection to scaling exponents

If we fit power laws of the form

$$L(N) \approx L_\infty + A_r N^{-\alpha_r}$$

in each phase, then under the toy assumptions we find:

- $\alpha_1 \approx 1$, with prefactor $A_1 \approx \lambda_1$;
- $\alpha_2 \approx 1$, with prefactor $A_2 \approx \lambda_2 > \lambda_1$.

On a log–log plot $L(N) - L_\infty$ vs N, the lines in the two regimes are roughly parallel (exponent ≈ -1) but with different vertical offsets due to different λ_r.

In realistic Transformers, the situation is more complex:

- exponents can deviate from 1 due to non-asymptotic effects;
- Fisher geometry may change gradually rather than via a single switch;
- multiple phased transitions can occur as more blocks and heads become active.

Still, the toy model illustrates how effective dimension changes—driven by singular geometry—create piecewise scaling behavior.

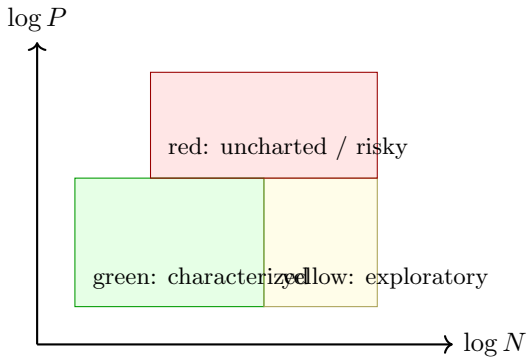

Figure 8: Safety windows on the scaling plane: regions marked "green" have well-understood scaling and geometry; "yellow" regions are exploratory but monitored; "red" regions are uncharted or high-risk and require special review.

9 Safety and Resource-Optimal Scaling Windows

Scaling experiments are resource-intensive and potentially risky: larger models can exhibit qualitatively new behaviors.

The singular scaling framework offers two complementary control levers:

- **Resource-optimality:** choosing (N, P, C) following compute-efficient scaling paths that yield maximal performance for a given budget.
- **Safety windows:** identifying regions where curvature, RLCT proxies, and capabilities behave in controlled ways.

OCTA Principle 9.1 (OCTA Scaling Principle IV – Safety windows on scaling surfaces). Define a *safety window* $\mathcal{W}$ on the scaling surface where:

- scaling exponents (α, β) are stable and well-characterized;
- RLCT proxies λ_{eff} and Fisher spectra remain within planned ranges;
- capability transitions are known and monitored;
- OCTA phase labels are in a set deemed acceptable (e.g. Phase 1 and early Phase 2).

OCTA experimentation should:

- expand $\mathcal{W}$ cautiously;
- avoid uncontrolled excursions into regions with unknown geometry or abrupt capability jumps.

In practice, this means:

- using the Scaling Surface Dashboard to visualize current and planned (N, P) locations;
- overlaying safety criteria to mark "green", "yellow", and "red" regions;
- gating large-scale experiments on explicit review of both performance and geometry metrics.

10 Synthesis with Days 1–4

Day 5 completes the first week's conceptual loop:

- **Day 1:** separated macro Singularity from micro singularities; scaling surfaces live in the macro story but are built from micro geometric features.
- **Day 2:** provided the attention-strata and block-cell view; scaling regimes correspond to changes in which strata are active, which heads are geometrically active, and how tie manifolds are traversed.
- **Day 3:** introduced SLT and RLCT; scaling exponents now have a principled connection to effective dimension, not raw parameter count.
- **Day 4:** gave Fisher-geometry probes and RLCT proxies; Day 5 uses these as conditioning signals for scaling experiments and regime detection.
- **Day 5:** elevates scaling laws to *singular scaling surfaces*, with geometry-aware exponents, capability loci, and OCTA phases.

From an OCTA perspective, scaling experiments are no longer just empirical curves; they are:

- probes of the singular geometry of Transformer function spaces;
- maps of capability transitions and safety windows;
- control interfaces for resource allocation, architecture design, and risk management.

Future days can move in two directions:

- deeper theoretical work on multi-parameter RLCT and its relation to scaling exponents and phase diagrams;
- concrete OCTA-scale experiments implementing D5.1–D5.5, generating real scaling surfaces with Fisher-conditioned interpretations and capability transition maps.

References

[1] S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford University Press, 2000.

[2] J. Kaplan, S. McCandlish, T. Henighan, et al. Scaling laws for neural language models. arXiv:2001.08361, 2020.

[3] S. Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge University Press, 2009.

[4] S. Watanabe. *Mathematical Theory of Bayesian Statistics*. CRC Press, 2018.