THE TWO REGIMES OF PRUNING UNDER ALEATORIC UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Nearly every machine learning problem grounded in the natural world exhibits irreducible uncertainty in its data-generating process, a property known as *aleatoric uncertainty*. Aleatoric uncertainty can leave the optimal decision rule unchanged, which raises a natural question: should the size of a neural network approximating this decision rule also remain unchanged? We study this question through *pruning*—the process of removing parameters from a model to obtain a smaller model—which we use as a proxy for the minimum model size needed to approximate the optimal decision rule. We find that pruned network size is not invariant to aleatoric uncertainty. Instead, sufficiently overparameterized neural networks exhibit two regimes of behavior: increasing aleatoric uncertainty initially increases pruned network size (the *expansion regime*), but beyond a critical point, further uncertainty decreases it (the *contraction regime*). Through ablations, we find that these two regimes arise from two competing effects: increasing aleatoric uncertainty increases the pruned network size required for a fixed performance target, while simultaneously lowering the performance target set by the dense model.

1 INTRODUCTION

Aleatoric uncertainty—the irreducible uncertainty inherent in a data-generating process—is woven into nearly every machine learning problem grounded in the natural world (Kendall & Gal, 2017; Depeweg et al., 2018). Measurements are imperfect, observations reveal only fragments of reality, and the same evidence can admit several reasonable interpretations. For example, different human annotators may classify the same image as either a dog or a cat (Schmarje et al., 2022). This irreducible label variability limits achievable task performance and, because it arises from the data-generating process itself, cannot be eliminated by collecting more data.

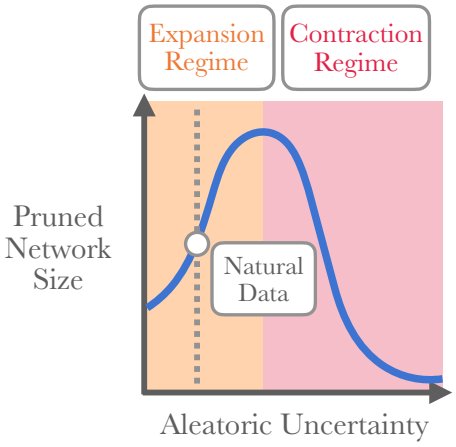

Figure 1: Depiction of the two regimes of pruning under aleatoric uncertainty. As aleatoric uncertainty increases from zero, the size of pruned networks increases (the expansion regime). However, beyond a critical point, increasing aleatoric uncertainty decreases pruned network size (the contraction regime).

While aleatoric uncertainty decreases achievable task performance, it need not change the optimal decision rule. If the optimal decision rule remains unchanged, one might therefore expect the model capacity required to approximate it to remain unchanged as well. We investigate this possibility through *pruning*, the process of removing parameters from a model to obtain a smaller model (Mozer & Smolensky, 1988; LeCun et al., 1989). We treat unpruned models as an empirical approximation to the optimal decision rule and use pruning to identify the smallest model that matches its performance (Arora et al., 2023). We call the size of this model the *pruned network size*. This framing leads to our central question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Is pruned network size invariant to aleatoric uncertainty?

We answer this question using label noise and diffusion to vary the uncertainty of MNIST and CIFAR-10 (Angluin & Laird, 1988; Ho et al., 2020; LeCun et al., 1998; Krizhevsky, 2009).

Our interventions reveal that pruned network size is not invariant to aleatoric uncertainty and instead produces two regimes of behavior. For sufficiently overparameterized models, pruned network size initially increases with aleatoric uncertainty (the *expansion regime*) until it reaches a critical point, then it begins to decrease (the *contraction regime*). As a corollary, we find that the naturally occurring aleatoric uncertainty in MNIST and CIFAR-10 contributes to larger pruned network sizes.

Our experiments distinguish two opposing effects of aleatoric uncertainty on pruned network size: *uncertainty overhead* and *target relaxation*. Uncertainty overhead is the increase in pruned network size required to achieve a fixed accuracy as uncertainty increases. Target relaxation arises because we determine the accuracy target from the dense model trained at each uncertainty level: when dense model accuracy decreases, the target falls, potentially allowing a smaller pruned network.

Target relaxation follows directly from our evaluation criterion: as aleatoric uncertainty lowers dense model accuracy, the accuracy target for the pruned network falls too. By contrast, our experiments do not isolate the mechanism underlying uncertainty overhead, which remains an open question.

To summarize, we present the following contributions in this paper:

- We empirically characterize uncertainty overhead, showing that it tends to increase with aleatoric uncertainty, target accuracy, and architecture size (Section 5).
- We show that target relaxation is significantly more pronounced for overparameterized models than for underparameterized models, although its dependence on model size is nonmonotonic (Section 6).
- For sufficiently overparameterized models, we identify two regimes of pruning induced by aleatoric uncertainty: an expansion regime and a contraction regime (Section 7).
- We develop a diffusion-based methodology for reducing aleatoric uncertainty in MNIST and CIFAR-10 and demonstrate that doing so decreases pruned network size (Section 8).

Aleatoric uncertainty is pervasive in natural datasets, yet it is typically fixed and often unmeasured, obscuring its influence on neural network behavior. Our results show that neural network compressibility depends not only on the decision rule being approximated, but also on the irreducible uncertainty of the data used to learn it, contributing to a broader understanding of how neural networks learn representations of data from the natural world.

2 Related Work

Our work touches on multiple topics in deep learning, including pruning and uncertainty estimation. See Appendix G for extended coverage of related work.

Uncertainty Estimation. Uncertainty estimation quantifies how uncertain a model is about its predictions, often by distinguishing uncertainty arising from inherent ambiguity in the data from uncertainty arising from limited knowledge of the model. Techniques for uncertainty estimation differ both in how they represent uncertainty and in how they summarize it. Bayesian neural networks approximate a posterior distribution over model parameters (Houlsby et al., 2011; Kendall & Gal, 2017), while deep ensembles use variability across independently trained models (Lakshminarayanan et al., 2017). In line with prior work, we use deep ensembles to estimate the aleatoric uncertainty of examples (Houlsby et al., 2011; Depeweg et al., 2018), to determine the aleatoric uncertainty of natural data distributions.

Pruning. Researchers have used pruning for interpretability, improving generalization, lowering memory footprints, and understanding training dynamics (Mozer & Smolensky, 1988; LeCun et al., 1989; Han et al., 2016; Frankle & Carbin, 2019; Frankle et al., 2020). Some work has found that noisy examples can impair the accuracy of pruned models (Jiang et al., 2024; Paul et al., 2022).

Complementary work shows that pruning disproportionately affects atypical, ambiguous, or mislabeled examples and can improve generalization by ceasing to fit noisy examples (Hooker et al., 2019; Jin et al., 2022). While these studies focus on accuracy and example-level behavior after pruning, our work studies the effect of aleatoric uncertainty on the size of pruned networks.

3 Formal Framework

The two regimes of pruning under aleatoric uncertainty concern the increase and decrease of pruned network size in response to aleatoric uncertainty, and this empirical pattern arises from a tension between uncertainty overhead and reference relaxation. We now develop the formal framework necessary to precisely quantify these concepts.

Data Distribution. Let P be the underlying data distribution and $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be observed data sampled from P. Inputs lie in $\mathcal{X}$, and labels lie in a finite set $\mathcal{Y}$. Write P_X for the marginal distribution of inputs and $p(y \mid x)$ for the conditional probability of label y given input x. We study classification: models are trained with cross-entropy loss, and their class decisions are obtained by taking the argmax of their probability predictions. A *Bayes decision rule* is a classifier $d\colon \mathcal{X} \to \mathcal{Y}$ satisfying $d(X) \in \arg\max_{y \in \mathcal{Y}} p(y \mid X)$ with probability one. A distribution may have multiple Bayes decision rules when labels tie for the highest conditional probability.

Aleatoric Uncertainty. The aleatoric uncertainty of a data distribution at a point x is the *Bayes-optimal risk*:

$$\mathcal{R}^\star(x) \coloneqq \inf_{q \in \mathcal{Q}} \mathbb{E}_{Y \sim p(\cdot|x)}[\ell(Y, q)],$$

where $\mathcal{Q}$ is the space of probability vectors over $\mathcal{Y}$ and ℓ is a loss function. For cross-entropy loss, $p(\cdot \mid x)$ minimizes the conditional loss, giving $\mathcal{R}^\star(x) = H(p(\cdot \mid x))$ (see Appendix A.1 for the derivation). We define the aleatoric uncertainty $\mathcal{R}^\star(P)$ of a data distribution as the expected conditional Bayes-optimal risk over inputs:

$$\mathcal{R}^\star(P) \coloneqq \mathbb{E}_{X \sim P_X}[\mathcal{R}^\star(X)].$$

Uncertainty Estimation. We estimate the entropy of the conditional label distribution $p(\cdot \mid x)$ with an ensemble of neural networks $g_1, \ldots, g_S : \mathcal{X} \to \mathcal{Q}$, where $g_j(x)$ is neural network j's predicted class distribution. We approximate the posterior expected aleatoric uncertainty by averaging the neural networks' conditional entropies:

$$\widehat{\mathcal{R}}^\star(x) \coloneqq \frac{1}{S} \sum_{j=1}^{S} H(g_j(x)) \approx \mathbb{E}_{G \sim \Pi_\mathcal{D}}[H(G(x))] = A(x; \Pi_\mathcal{D}), \tag{1}$$

where $\Pi_\mathcal{D}$ is a posterior distribution over probabilistic predictors given $\mathcal{D}$, and $A(x; \Pi_\mathcal{D})$ is the posterior expected aleatoric uncertainty (see Appendix A.1.2 for more details).

Uncertainty Modifiers. To study how aleatoric uncertainty affects pruning, we modulate aleatoric uncertainty while preserving the set of Bayes decision rules. We formalize such a transformation as an *uncertainty modifier* Φ_α, which maps P to a new data distribution satisfying $\mathcal{R}^\star(\Phi_\alpha(P)) = \alpha$. We use two uncertainty modifiers: *uniform label noise* and *distribution shaping*.

4 Experimental Setup

We briefly describe the methodology common to all of our experiments. See Appendices A, B, and E for details.

Dataset Targets. As sources of natural data, we evaluate on MNIST and CIFAR-10 (LeCun et al., 1998; Krizhevsky, 2009), which are both common computer vision benchmarks.

Training and Pruning. Within each student-training setting, we hold the architecture, optimizer, and learning-rate schedule fixed across uncertainty levels; these settings use AdamW with cosine learning rate decay. We train a dense model, then apply global iterative magnitude pruning with learning-rate rewinding, removing 20% of the remaining kernel weights per round while leaving biases unpruned. We use three model seeds per experimental condition; generated-data experiments cross these with three dataset replicas, giving nine runs per condition.

For MNIST, we use MLPs with two hidden layers and hidden widths varying in $\{32, 64, 128, 256, 512, 1024, 2048, 4096\}$. For CIFAR-10, we use the WideResNet (WRN) architecture (Zagoruyko & Komodakis, 2016), with the configuration WRN-28-k, where 28 is the network depth and the widen factor k varies over $\{0.5, 1, 2, 4, 8\}$. Both architecture grids include sizes below and above the interpolation threshold n_0 for the unmodified datasets.

Determining Pruned Network Size. In principle, one can obtain any sparsity using iterative magnitude pruning by adjusting the percentage of remaining weights to prune in each iteration. Thus, to determine pruned network size at a target test accuracy, we linearly interpolate test accuracy and retained parameter count between adjacent checkpoints within each run's pruning trajectory. Then, we select the smallest parameter count among qualifying measured checkpoints and interpolated threshold crossings. We perform this selection separately for each run before aggregating results.

An Ensemble Estimate for Aleatoric Uncertainty. We construct the estimate for (1) as follows. For each dataset, we train five independently initialized classifiers for each of five data folds, and we estimate the uncertainty of an example only with classifiers trained on folds that excluded that example. We calibrate predicted class probabilities using a single shared temperature fitted to minimize the negative log-likelihood of the ensemble's out-of-fold predictions.

5 Measuring Uncertainty Overhead

Increasing aleatoric uncertainty can both increase the pruned network size needed to achieve a fixed accuracy and lower the accuracy target set by the dense model. To isolate uncertainty overhead from target relaxation, we compare pruned network sizes across aleatoric uncertainty levels while holding the target test accuracy fixed. We repeat this comparison across architecture sizes and accuracy targets to characterize how uncertainty overhead depends on each.

Methodology. Uniform label noise is an uncertainty modifier that interpolates between the original labels and a uniform distribution over all labels. Concretely, uniform label noise on $K \geq 2$ classes replaces each label with a uniformly sampled different label with probability η, yielding P_η with unchanged P_X and conditional probabilities

$$p_\eta(y \mid x) = (1 - K\eta/(K-1))\, p(y \mid x) + \eta/(K-1), \qquad 0 \leq \eta < (K-1)/K.$$

The positive coefficient preserves the set of Bayes decision rules.[1]

To implement uniform label noise on a training set with N examples, we select $\lfloor \eta N \rfloor$ examples uniformly at random without replacement, where $\eta \in [0, 1]$ is the requested corruption fraction. We evaluate corruption percentages $\eta \in \{0\%, 1\%, 2\%, 3\%, 5\%, 10\%, 20\%, 30\%, 50\%\}$, all of which preserve the set of Bayes decision rules. We corrupt only the training labels, and we compute the expected corrupted test accuracy as a function of model performance on the uncorrupted test set. We estimate the aleatoric uncertainty after corruption by applying the relabeling rule to each uncertainty-ensemble classifier's predicted class probabilities and averaging their entropies over training inputs. For each architecture and corruption fraction, we use the mean test accuracy minus one standard deviation as the target test accuracy for determining the pruned network size.

For each architecture, we measure uncertainty overhead as the signed difference in pruned network size required to achieve the same test accuracy on the original test set:

$$U_\alpha(t) = M_\alpha(t) - M_{\alpha_0}(t),$$

[1] At $\eta = (K-1)/K$, the conditional label distribution is uniform. Above this threshold, class rankings reverse, so the set of Bayes decision rules need not be preserved. If $\mathcal{R}^\star(P) < \log K$, strict concavity makes mean entropy increase continuously and strictly toward $\log K$. Each target $\alpha \in [\mathcal{R}^\star(P), \log K)$ then determines a unique η and uncertainty modifier $\Phi_\alpha(P) = P_\eta$.

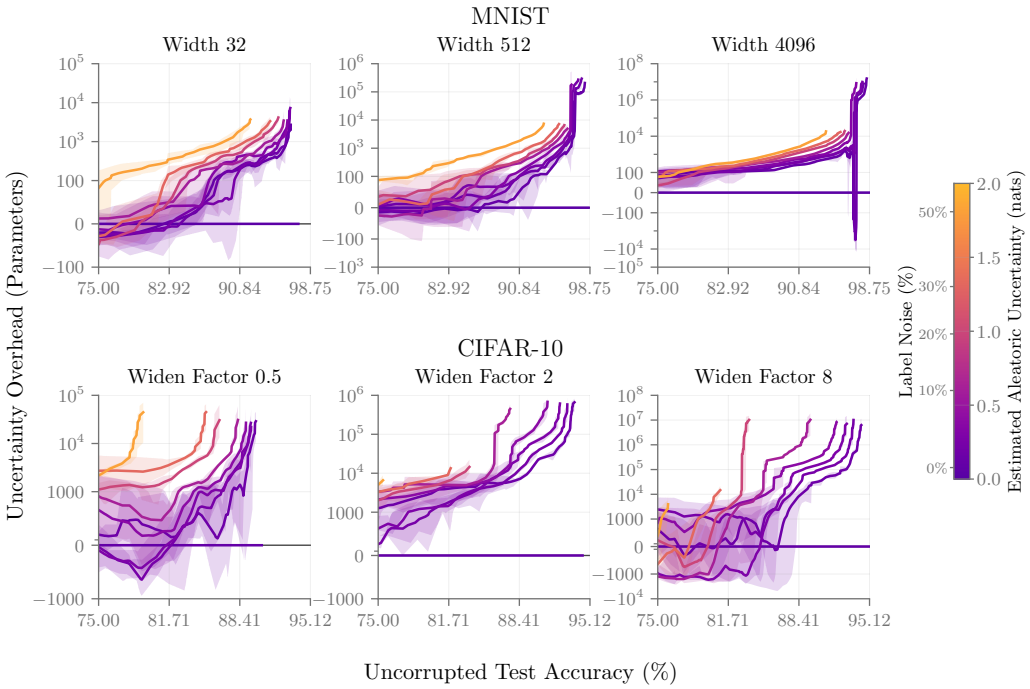

Figure 2: For a fixed test accuracy, increasing aleatoric uncertainty tends to increase uncertainty overhead, which identifies a source of the increase in pruned network size in the expansion regime. Curves show averages, shaded bands show min–max ranges, and color indicates training aleatoric uncertainty. The test accuracy axis is linear, and the uncertainty overhead axis is symmetric-log with linear regions of ± 100 parameters for MNIST and ± 1000 for CIFAR-10.

where $M_\alpha(t)$ is the minimal retained parameter count needed to achieve test accuracy at least t under our training and pruning procedure on a dataset with aleatoric uncertainty α. Positive values indicate that training on the modified dataset requires a larger pruned network to meet the same accuracy target. We estimate these differences from paired pruning trajectories, using interpolation to determine pruned network size and restricting comparisons to targets supported by both conditions.

See Appendix B for details of label corruption, aleatoric uncertainty estimation, dataset splits, and the model training and pruning protocols.

Results. Figure 2 shows uncertainty overhead as a function of test accuracy for MNIST and CIFAR-10. Uncertainty overhead occurs at every evaluated architecture size, but its magnitude depends strongly on the target accuracy and aleatoric uncertainty. Holding the target accuracy and architecture size fixed, uncertainty overhead generally increases as aleatoric uncertainty increases. Holding the aleatoric uncertainty and architecture size fixed, uncertainty overhead also generally increases with the target accuracy. Holding the target accuracy and aleatoric uncertainty fixed, uncertainty overhead generally increases with architecture size (see Figures 8 and 9 in Appendix C for figures that provide a clearer view of this property). Thus, uncertainty overhead tends to increase with aleatoric uncertainty, test accuracy, and architecture size.

6 Measuring Target Relaxation

Increasing aleatoric uncertainty can lower dense model accuracy, relaxing the accuracy requirement used to determine pruned network size. To measure target relaxation, we compare the accuracy targets set by dense models trained at different aleatoric uncertainty levels. We repeat this comparison across architecture sizes to characterize how target relaxation depends on initial model size.

Methodology. We use the same uniform label noise methodology as in Section 5.

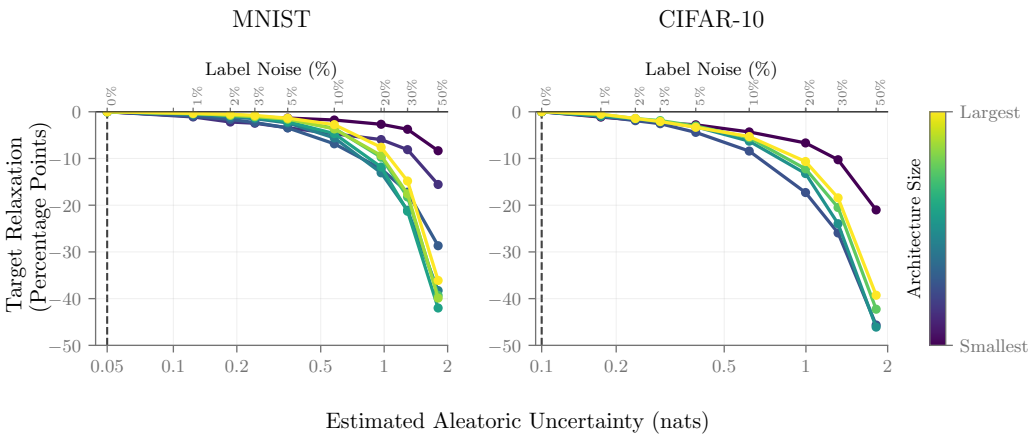

Figure 3: Overparameterized models tend to exhibit more target relaxation than underparameterized models, which identifies a source of the decrease in pruned network size in the contraction regime. Curves fix architecture size. The shared colorbar orders architectures from smallest to largest within each dataset. Negative values indicate that the performance requirement has relaxed relative to zero label corruption. Dashed vertical lines mark the estimated uncertainty of the original datasets.

For each architecture, we measure target relaxation as the signed change in the test accuracy threshold that determines pruned network size:

$$R_\alpha = \tau_\alpha - \tau_{\alpha_0}.$$

Here, $\tau_\alpha = \mu_\alpha - \sigma_\alpha$, where μ_α and σ_α are the mean and population standard deviation of test accuracy across the three dense models trained on a dataset with aleatoric uncertainty α, and we use the original, uncorrupted test set. Negative values indicate a lower accuracy threshold for models trained on the modified dataset, with more negative values indicating stronger target relaxation.

Results. Figure 3 shows target relaxation as a function of aleatoric uncertainty for MNIST and CIFAR-10. Every evaluated architecture exhibits target relaxation, and target relaxation increases as aleatoric uncertainty increases. For both datasets, the largest architecture shows more target relaxation than the smallest architecture at label noises greater than 5%. However, the trend between these architecture sizes is not monotone; as a function of architecture size, target relaxation becomes stronger then becomes weaker. At the highest aleatoric uncertainty, only MNIST MLP widths $\geq$512 and CIFAR-10 widen factors $\geq$2 achieve $\geq 99\%$ training accuracy. Increasing architecture size beyond these sizes begins to decrease target relaxation. This result runs counter to prior work by Xue et al. (2024), which shows that in the overparameterized regime and at high levels of label noise, increasing parameters can worsen test accuracy. These results suggest that pruned network size decreases more sharply for overparameterized models than it does for underparameterized models. See Appendix C.2 for the underlying dense test accuracies.

7 The Two Regimes of Pruning Under Aleatoric Uncertainty

Having measured uncertainty overhead and target relaxation separately, we now examine their combined effect on pruned network size. We use label noise to increase aleatoric uncertainty in MNIST and CIFAR-10 and measure pruned network size against the accuracy target set at each aleatoric uncertainty level. We repeat this comparison across architecture sizes to identify when increasing aleatoric uncertainty produces expansion followed by contraction.

Methodology. We use the same uniform label noise methodology as in Section 5.

For each architecture and aleatoric uncertainty level α, we set the target test accuracy to $\tau_\alpha = \mu_\alpha - \sigma_\alpha$, where μ_α and σ_α are the mean and standard deviation of test accuracy across the dense models trained at that uncertainty level. We compute pruned network size as the smallest parameter

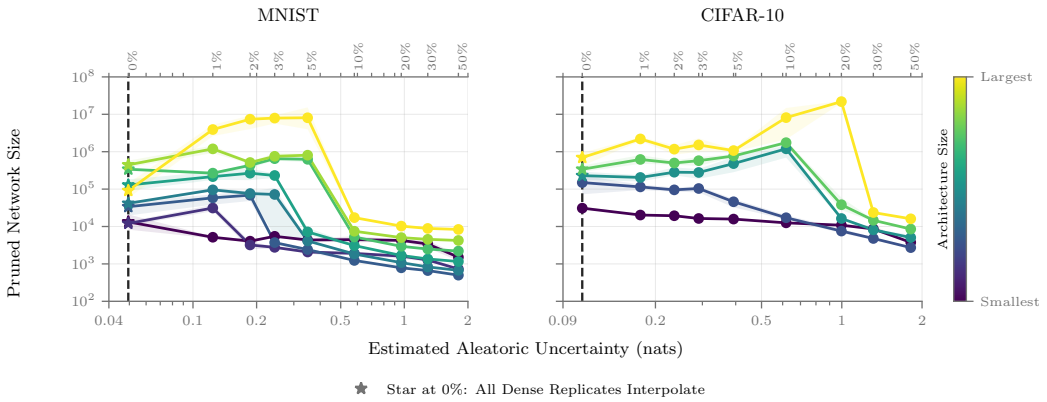

Figure 4: For sufficiently overparameterized models, increasing aleatoric uncertainty increases pruned network size, then as aleatoric uncertainty continues increasing, pruned network size begins to decrease. The horizontal axes use a logarithmic scale; the lower axes show estimated aleatoric uncertainty and the upper axes show the label noise percentage. Each curve corresponds to an architecture size. Dashed vertical black lines mark the estimated uncertainty of the original MNIST and CIFAR-10 datasets, respectively. Shaded areas above and below points on a curve show the min and max pruned network size across seeds. Stars at 0% label noise indicate that all dense replicates interpolate the training data.

count among pruning checkpoints exceeding τ_α and interpolated threshold crossings.Because the accuracy target is determined separately at each aleatoric uncertainty level, these measurements capture the combined effects of uncertainty overhead and target relaxation.

Results. Figure 4 shows the results of increasing aleatoric uncertainty on natural data. See Appendix C for figures showing the underlying pruning trajectories.

For all MNIST MLP widths ≥ 64 and CIFAR-10 widen factors ≥ 2, we observe an increase in pruned network size for at least one aleatoric uncertainty, relative to the original dataset's aleatoric uncertainty. All architecture sizes below these bounds show decreases in pruned network size, relative to the original dataset's aleatoric uncertainty. These results demonstrate that MNIST and CIFAR-10 lie in the expansion regime at their observed uncertainty levels. For the architecture sizes that show an increase in pruned network size, the label noise level where they achieve their maximum is from 1–5% (0.12–0.35 nats) for MNIST and 10–20% (0.62–1.0 nats) for CIFAR-10. The trend for the label noise that induces the maximum is not monotone in architecture size.

Every network that initially shows an increase in pruned network size as aleatoric uncertainty increases subsequently shows a steep drop in pruned network size as aleatoric uncertainty continues to increase. As the architecture size becomes larger, the aleatoric uncertainty at which we see decreases in pruned network size increases.

These results demonstrate the existence of both an expansion regime and a contraction regime on these datasets and sufficiently large instantiations of these model architectures.

Why Highly Overparameterized Models? Our results in Sections 5 and 6 suggest an explanation why highly overparameterized models tend to exhibit greater expansion and contraction than smaller models. Figure 5 shows the underlying dense-accuracy curves, which show that larger models achieve higher test accuracy than smaller models at low aleatoric uncertainty, and the results in Section 5 show that higher accuracy targets show greater uncertainty overhead. Thus, at low aleatoric uncertainty, the stronger targets imposed by larger models can expose greater uncertainty overhead and favor expansion. At high aleatoric uncertainty, their larger target reductions can allow target relaxation to outweigh uncertainty overhead and favor contraction. By comparison, smaller models start with lower accuracy targets, but their accuracies do not drop as precipitously as for larger models. Thus, even though smaller models also exhibit uncertainty overhead at higher accuracy targets, the low accuracy targets they set tend not to elicit this uncertainty overhead and hence do not generate as much expansion.

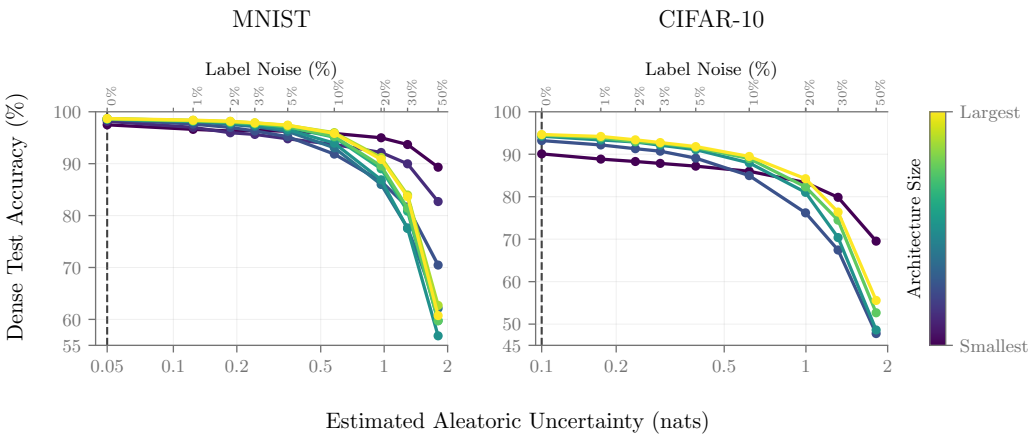

Figure 5: Dense test accuracy as a function of estimated conditional entropy for MNIST uniform corruption (left) and CIFAR-10 uniform corruption (right). Curves fix architecture size; lines are means and bands span seed minima and maxima. The shared colorbar encodes architecture size, from the smallest to the largest MNIST hidden width or CIFAR-10 widen factor.

8 Reducing the Aleatoric Uncertainty of Natural Data

We used uniform label noise to observe the transition from the expansion regime to the contraction regime. However, label noise can only increase aleatoric uncertainty. We now use *distribution shaping* to reduce the aleatoric uncertainty of MNIST and CIFAR-10 and evaluate whether this intervention reduces pruned network size.

8.1 Methodology

We describe how we implement distribution shaping and the target accuracy we use for pruned network size in these experiments. See Appendix E for more detailed methodology.

Distribution Shaping. Distribution shaping is an uncertainty transformer that reweights the input distribution while keeping $p(\cdot \mid x)$ fixed, preserving the set of Bayes decision rules with probability one. We use exponential tilting, which defines P^λ by reweighting P_X proportionally to $e^{\lambda h(x)}$ for $h(x) = H(p(\cdot \mid x))$, keeping $p(\cdot \mid x)$ fixed.[2] An important property of exponential tilting is that, when it achieves a target aleatoric uncertainty, no other distribution with that uncertainty is closer to the reference in forward KL divergence (see Appendix E.7 for a proof).

For each of MNIST and CIFAR-10, we first estimate the aleatoric uncertainty of each original training example. We group these estimates by class label and construct a separate histogram for each class using bins of width 0.025 nats, so we can ultimately construct class-balanced datasets.

To realize these target histograms, we generate a shared pool of candidate images for each dataset using class-conditional diffusion models, then select images such that we exactly match the target histogram. For MNIST, we train a class-conditional diffusion model following the methodology of Karras et al. (2022); for CIFAR-10, we use their released model weights. We construct three mutually disjoint dataset replicas per target aleatoric uncertainty. Our main comparison evaluates two aleatoric uncertainty targets for each dataset: the lowest selected target supported by the image pool we generated and the target corresponding to the original dataset's aleatoric uncertainty. In principle, one could apply distribution shaping to increase the aleatoric uncertainty of a data distribution as well. However, the maximum aleatoric uncertainty afforded by the image pools we generated is only modestly larger ($0.056 \to 0.065$ nats for MNIST and $0.11 \to 0.12$ nats for CIFAR-10). For simplicity, we only use this methodology to reduce aleatoric uncertainty.

[2]Let h_- and h_+ be the tightest bounds satisfying $h_- \leq h(X) \leq h_+$ with probability one. If $h_- < h_+$, each target $\alpha \in (h_-, h_+)$ determines a unique tilting parameter $\lambda \in \mathbb{R}$ and uncertainty modifier $\Phi_\alpha(P) = P^\lambda$. Reweighting keeps mean entropy in $[h_-, h_+]$.

Table 1: Pruned network size at the original and lowest selected aleatoric uncertainty targets for MNIST (left; 0.06 and 0.02 nats) and CIFAR-10 (right; 0.11 and 0.03 nats). Sizes are nine-run means ± sample standard deviations across three dataset replicas and three model seeds; the k suffix denotes thousands of parameters. Percentage reductions are computed from the unrounded means.

MNIST

Hidden Width	Original Size	Reduced Size	Reduction
32	6k ± 2k	4k ± 1k	43%
64	13k ± 8k	7k ± 2k	50%
128	19k ± 4k	11k ± 3k	45%
256	28k ± 6k	14k ± 2k	52%
512	121k ± 25k	20k ± 10k	83%
1024	475k ± 484k	21k ± 8k	95%
2048	379k ± 97k	20k ± 10k	95%
4096	134k ± 164k	57k ± 18k	58%

CIFAR-10

Widen Factor	Original Size	Reduced Size	Reduction
0.5	24k ± 3k	26k ± 2k	−8%
1	69k ± 9k	63k ± 11k	9%
2	127k ± 25k	99k ± 32k	22%
4	299k ± 121k	226k ± 44k	24%
8	600k ± 215k	425k ± 70k	29%

Fixed Matching Criterion. In Section 7, we subtract the standard deviation from the mean dense test accuracy to set the target accuracy for determining the pruned network size. In this experiment, as aleatoric uncertainty decreases, dense test accuracy approaches 100% and the standard deviation contracts (see Appendix F.6). Consequently, a comparison under this criterion mixes changes in pruned network size with a systematic change in criterion strictness. Instead, for architecture size n, we compute the standard deviation $\sigma_{\text{dense}}(\mu_{\text{orig}}, n)$ at the original dataset's aleatoric uncertainty μ_{orig} and reuse it at every target μ.

8.2 Results

Table 1 compares pruned network size at the original and lowest selected aleatoric uncertainty targets for MNIST and CIFAR-10. The full set of target levels and results appears in Appendix F.1.

For all eight MNIST widths, the pruned network size is 43%–95% smaller at the lowest aleatoric uncertainty than at the original aleatoric uncertainty. For CIFAR-10 widen factors 1, 2, 4, and 8, the pruned network size is 9%–29% smaller at the lowest aleatoric uncertainty than at the original aleatoric uncertainty; widen factor 0.5 instead increases by 8%.

Transfer to the Original Distribution. Distribution shaping changes the input marginal, so its smaller pruned network sizes could reflect an easier training distribution. We test whether models pruned on data with reduced aleatoric uncertainty can adapt to the original distribution by finetuning them. We find that finetuning these models largely closes the accuracy gap and can even improve over models pruned on the original data (see Appendix F.5). These results suggest that our methodology does not significantly decrease task complexity.

9 Conclusion

An unchanged optimal decision rule need not imply an unchanged pruned network size. Through controlled experiments on MNIST and CIFAR-10, we identify two regimes in sufficiently overparameterized networks: increasing aleatoric uncertainty first increases pruned network size, then decreases it. Our analysis distinguishes two competing effects underlying this pattern: uncertainty overhead, which increases the size required to achieve a fixed accuracy, and target relaxation, which lowers the accuracy requirement as dense-model performance declines.

A central question remains: why does uncertainty overhead arise, and to what extent can training and pruning methods avoid it? Our experiments characterize this overhead without isolating its mechanism, leaving an opportunity to distinguish the capacity a task intrinsically requires from the capacity our learning and pruning procedures retain. Understanding uncertainty overhead could clarify how neural networks learn from natural data with aleatoric uncertainty.

AI USE STATEMENT

We used generative AI tools to assist with proposing and refining hypotheses, developing theoretical arguments and proofs, designing experiments, processing data, implementing and running experiment code, interpreting results, conducting literature review, and writing portions of this paper. The authors reviewed AI-assisted work by reading the agents' outputs and written contributions and questioning the agents about what they had done. The authors take responsibility for the final content of this paper, including its text, claims, and artifacts produced with generative AI assistance.

REPRODUCIBILITY STATEMENT

Section 4 summarizes the experimental setup, and Appendix A provides the uncertainty definitions and calibrated ensemble protocol. Appendices B and E specify dataset preparation and splits, label corruption, generated-data construction, model architectures, hyperparameters, training and pruning schedules, checkpoint selection, and aggregation across model seeds and dataset replicas. The endpoint-finetuning protocol is described in Appendix E.6, and Appendix D reports checks of architecture-size, pruning-rule, and checkpoint-selection sensitivity. The assumptions, derivations, and proofs supporting our analysis are provided in Appendices A.1, B.2, and E.7.

REFERENCES

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 1988.

Zachary Ankner, Alex Renda, Gintare Karolina Dziugaite, Jonathan Frankle, and Tian Jin. The effect of data dimensionality on neural network prunability, 2022.

Viplove Arora, Daniele Irto, Sebastian Goldt, and Guido Sanguinetti. Quantifying lottery tickets under label noise: accuracy, calibration, and complexity. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2023.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. doi: 10.1073/pnas.1907378117.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems*, volume 2, 2020. URL https://proceedings.mlsys.org/paper/2020/hash/6c44dc73014d66ba49b28d483a8f8b0d-Abstract.html.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

Xiangyu Chang, Yingcong Li, Samet Oymak, and Christos Thrampoulidis. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. In *AAAI Conference on Artificial Intelligence*, 2021.

Imre Csiszár. i-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975. doi: 10.1214/aop/1176996454.

Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004. doi: 10.1561/0100000004.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1184–1193, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Spencer Frei, Niladri S. Chatterji, and Peter Bartlett. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. In *Proceedings of the Thirty-Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 2668–2703, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

Zhengqi He, Zeke Xie, Quanzhi Zhu, and Zengchang Qin. Sparse double descent: Where network pruning aggravates overfitting. In *International Conference on Machine Learning*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021. URL `https://www.jmlr.org/papers/v22/21-0366.html`.

Sara Hooker, Aaron C. Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget?, 2019.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011.

Runqing Jiang, Yan Yan, Jing-Hao Xue, Biao Wang, and Hanzi Wang. When sparse neural network meets label noise learning: A multistage learning framework. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2208–2222, 2024. doi: 10.1109/TNNLS.2022.3188799.

Shuwen Jin, Junzhu Mao, Zeren Sun, and Yazhou Yao. Discriminative response pruning for robust and efficient deep networks under label noise. *Pattern Recognition Letters*, 2026.

Tian Jin, Michael Carbin, Daniel M. Roy, Jonathan Frankle, and Gintare Karolina Dziugaite. Pruning's effect on generalization through the lens of training and regularization. In *Advances in Neural Information Processing Systems*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1137–1145, 1995.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lorenz Kuhn, Clare Lyle, Aidan N. Gomez, Jonas Rothfuss, and Yarin Gal. Robustness to pruning predicts generalization in deep neural networks, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems*, 2017.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, 1989.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J. Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23536–23557, 2023.

Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in Neural Information Processing Systems*, 1988.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 2021.

Mansheej Paul, Brett W. Larsen, Surya Ganguli, Jonathan Frankle, and Gintare Karolina Dziugaite. Lottery tickets on a data diet: Finding initializations with sparse trainable networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18916–18928, 2022.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9617–9626, 2019.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Victor Quétu and Enzo Tartaglione. DSD^2: Can we dodge sparse double descent and compress the neural network worry-free? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024. doi: 10.1609/aaai.v38i13.29393.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020.

Lars Schmarje, Vasco Grossmann, Claudius Zelenka, Sabine Dippel, Rainer Kiko, Mariusz Oszust, Matti Pastell, Jenny Stracke, Anna Valros, Nina Volkmann, and Reinhard Koch. Is one annotation enough? a data-centric image classification benchmark for noisy and ambiguous label estimation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33215–33232, 2022.

Ilia Sucholutsky, Ruairidh M. Battleday, Katherine M. Collins, Raja Marjieh, Joshua Peterson, Pulkit Singh, Umang Bhatt, Nori Jacoby, Adrian Weller, and Thomas L. Griffiths. On the informativeness of supervision signals. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 2036–2046, 2023.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9275–9293, 2020. doi: 10.18653/v1/2020.emnlp-main.746.

Nuri Mert Vural and Murat A. Erdogdu. Pruning is optimal for learning sparse features in high-dimensions. In *Proceedings of the Thirty-Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 4787–4861, 2024.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 2282–2292, 2023.

Yihao Xue, Kyle Whitecross, and Baharan Mirzasoleiman. Investigating the impact of model width and density on generalization in presence of label noise. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, 2024.

Hongru Yang, Yingbin Liang, Xiaojie Guo, Lingfei Wu, and Zhangyang Wang. Random pruning over-parameterized neural networks can improve generalization: A training dynamics analysis. *Journal of Machine Learning Research*, 26(84):1–51, 2025.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Yefan Zhou, Yaoqing Yang, Arin Chang, and Michael W. Mahoney. A three-regime model of network pruning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42790–42809, 2023.

A Shared Experimental Methods

We first give the uncertainty definitions and calibrated ensemble protocol shared by the label-noise and distribution-shaping experiments. Appendices B and E specify the intervention-specific data splits, student training, pruning, and evaluation procedures.

A.1 Aleatoric Uncertainty Derivations

We derive the pointwise Bayes-optimal risk and the Bayesian interpretation of the aleatoric uncertainty estimator introduced in Section 3.

A.1.1 CLASSIFICATION

Let $\mathcal{Y}$ be a finite set, let predictions be distributions $q \in \mathcal{Q} = \Delta(\mathcal{Y})$, and let $\ell(y, q) = -\log q(y)$ be the per-example cross-entropy loss. Then

$$\begin{aligned}
\mathcal{R}^\star(x) &= \inf_{q \in \Delta(\mathcal{Y})} \mathbb{E}_{Y \sim p_P(\cdot|x)}[-\log q(Y)] \\
&= \inf_{q \in \Delta(\mathcal{Y})} \mathbb{E}_P[-\log q(Y) \mid X = x] \\
&= \inf_{q \in \Delta(\mathcal{Y})} \sum_{y \in \mathcal{Y}} p_P(y \mid x)(-\log q(y)) \\
&= \inf_{q \in \Delta(\mathcal{Y})} \sum_{y \in \mathcal{Y}} p_P(y \mid x)(-\log p_P(y \mid x)) \\
&\quad + \sum_{y \in \mathcal{Y}} p_P(y \mid x) \log \frac{p_P(y \mid x)}{q(y)} \\
&= \inf_{q \in \Delta(\mathcal{Y})} H(p_P(\cdot \mid x)) + \mathrm{KL}(p_P(\cdot \mid x) \,\|\, q) \\
&= H(p_P(\cdot \mid x))
\end{aligned}$$

Thus, with cross-entropy loss, the pointwise aleatoric uncertainty is the conditional-label entropy. Let $\mathcal{B}_P(x) \coloneqq \arg\max_{y \in \mathcal{Y}} p(y \mid x)$ denote the labels with maximum conditional probability at x. For the modifiers considered in Section 3, the set of Bayes decision rules is preserved exactly when $\mathcal{B}_{\Phi_\alpha(P)}(X) = \mathcal{B}_P(X)$ with probability one.

A.1.2 BAYESIAN ESTIMATION OF ALEATORIC UNCERTAINTY

To estimate the aleatoric uncertainty of data, we connect $\mathcal{R}^\star(x) = H(p(\cdot \mid x))$ to a Bayesian decomposition of predictive uncertainty. Let $\Pi_\mathcal{D}$ be a posterior distribution over probabilistic predictors $g \colon \mathcal{X} \to \mathcal{Q}$ given $\mathcal{D}$. Define the posterior predictive distribution and the posterior expected aleatoric uncertainty, respectively, as

$$\bar{p}_\mathcal{D}(\cdot \mid x) \coloneqq \mathbb{E}_{G \sim \Pi_\mathcal{D}}[G(x)], \qquad A(x; \Pi_\mathcal{D}) \coloneqq \mathbb{E}_{G \sim \Pi_\mathcal{D}}[H(G(x))],$$

where $G \sim \Pi_\mathcal{D}$ is the predictive model and $\widetilde{Y} \mid G, x \sim G(x)$ is the predictive distribution. The predictive entropy decomposes as (Depeweg et al., 2018)

$$\underbrace{H(\bar{p}_\mathcal{D}(\cdot \mid x))}_{\text{total predictive entropy}} = \underbrace{A(x; \Pi_\mathcal{D})}_{\text{posterior expected aleatoric uncertainty}} + \underbrace{I(\widetilde{Y}; G \mid x, \mathcal{D})}_{\text{mutual information}}.$$

The quantity $A(x; \Pi_\mathcal{D})$ is a posterior estimate of the data distribution's aleatoric uncertainty $\mathcal{R}^\star(x)$. If $\Pi_\mathcal{D}$ places all of its mass on the true conditional predictor $g_P(x) = p(\cdot \mid x)$, then $A(x; \Pi_\mathcal{D}) = \mathcal{R}^\star(x)$ and the mutual information vanishes.

In our experiments, we approximate $\Pi_\mathcal{D}$ by a uniform distribution over an ensemble of neural networks. This posterior approximation is heuristic, and ensemble-averaged entropy need not equal the true aleatoric uncertainty (Wimmer et al., 2023).

A.2 UNCERTAINTY ENSEMBLE PROTOCOL

Cross-Fitting and Model Selection. We estimate each original training example's class probabilities using teachers that were not trained on that example, since in-sample confidence can reflect memorization rather than label quality; related label-auditing methods likewise rely on out-of-sample predicted probabilities (Northcutt et al., 2021). For each dataset, we partition the official training set $\mathcal{D}$ into $K = 5$ stratified outer folds $F_1, \ldots, F_K$, preserving the class proportions within each fold, as recommended for classification error estimation (Kohavi, 1995). For outer fold k, we reserve a separate stratified validation set V_k of 5,000 examples from $\mathcal{D} \setminus F_k$ and optimize the teacher only on $\mathcal{D} \setminus (F_k \cup V_k)$. Thus each MNIST teacher is optimized on 43,000 examples and each CIFAR-10 teacher on 35,000 examples. We select the checkpoint with the highest accuracy on V_k, breaking ties by lower validation cross-entropy. Separating V_k from the held-out fold ensures

that F_k affects neither weight optimization nor checkpoint selection, avoiding the selection-induced optimism that can arise when evaluation data influence model selection (Cawley & Talbot, 2010).

For each outer fold, we train $S = 5$ teachers from independent random initializations and data-order seeds. Multiple independently trained networks provide a simple and empirically effective way to obtain both more stable predictions and a measure of predictive disagreement (Lakshminarayanan et al., 2017). The choices $K = S = 5$ give each example predictions from five independently trained models while keeping the procedure tractable: in total, the method trains 25 teachers per dataset. For an example $x_i \in F_k$, we retain only the logits $z_{i,s} \in \mathbb{R}^{10}$ produced by the five teachers associated with fold k. Consequently, every training example receives strictly out-of-fold logits.

Teacher Models. For MNIST, each teacher is a ReLU multilayer perceptron with widths $784 \to 4096 \to 4096 \to 10$. Inputs are scaled to $[0, 1]$ without augmentation, and the network is trained on the original hard labels for 200 epochs using AdamW (Loshchilov & Hutter, 2019), a batch size of 256, an initial learning rate of 3×10^{-4}, and cosine learning-rate decay. For CIFAR-10, each teacher is a Wide ResNet-28-10 (Zagoruyko & Komodakis, 2016). We apply per-channel normalization, random horizontal flips, and random 32×32 crops after four-pixel padding. The network is trained for 200 epochs with stochastic gradient descent, batch size 128, initial learning rate 0.1, momentum 0.9, weight decay 5×10^{-4}, and dropout 0.3; the learning rate is reduced by a factor of ten after epochs 60, 120, and 160.

Out-of-Fold Calibration and Ensembling. Modern neural classifiers can be systematically overconfident, and post-hoc temperature scaling is a strong low-capacity calibration baseline (Guo et al., 2017). For a candidate temperature $T > 0$, we convert each out-of-fold logit vector to

$$p_{i,s}^{(T)}(c) = \frac{\exp(z_{i,s,c}/T)}{\sum_{c'=1}^{10} \exp(z_{i,s,c'}/T)}. \tag{2}$$

We fit one temperature shared by all folds and ensemble members by minimizing the negative log-likelihood of the probability-averaged out-of-fold ensemble,

$$T^* = \underset{T \in [0.05, 20]}{\arg\min} \; -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log \left[\frac{1}{S} \sum_{s=1}^{S} p_{i,s}^{(T)}(y_i) \right]. \tag{3}$$

The predictive logits entering this optimization are out of fold; the only quantity fitted using all training labels is the single global scalar T^*. Using one scalar limits calibration capacity and puts the independently trained teachers on a common confidence scale.

The calibrated ensemble prediction for example i is the probability mixture

$$q_i(c) = \frac{1}{S} \sum_{s=1}^{S} p_{i,s}^{(T^*)}(c). \tag{4}$$

Example-Level Aleatoric Uncertainty Estimate. Following the uncertainty decomposition in Appendix A.1.2, we estimate aleatoric uncertainty with the mean entropy of the calibrated members,

$$H_{\exp}(i) = \frac{1}{S} \sum_{s=1}^{S} \left[-\sum_{c} p_{i,s}^{(T^*)}(c) \log p_{i,s}^{(T^*)}(c) \right]. \tag{5}$$

This estimate is high when individual teachers assign diffuse class probabilities rather than merely disagreeing with one another. Independently trained teachers are not exact posterior samples, so this model-based estimate need not uniquely or perfectly isolate the data's aleatoric uncertainty (Wimmer et al., 2023). The label-noise experiments apply the corruption transformation to each member's calibrated probabilities before taking its entropy; the generated-data experiments use these estimates to define selection histograms.

B Uniform Label-Noise Methodology

This appendix gives the corruption, training, pruning, and evaluation procedures for the MNIST and CIFAR-10 experiments in Sections 5, 6, and 7. Appendix A.2 describes the shared calibrated ensemble used to estimate aleatoric uncertainty.

B.1 Uniform Label-Noise Intervention

In classification, aleatoric uncertainty specializes to conditional entropy. We vary it with uniform forced-different label noise on MNIST and CIFAR-10. The reported uniform-noise grid is

$$\eta \in \{0\%, 1\%, 2\%, 3\%, 5\%, 10\%, 20\%, 30\%, 50\%\}.$$

The intervention leaves the input marginal and the validation and test sets unchanged; only training labels are corrupted.

Uniform forced-different label noise. For a K-class training set of size N, we select exactly $m = \lfloor \eta N \rfloor$ examples uniformly without replacement. Each selected label is replaced by a class drawn uniformly from the other $K - 1$ classes. Thus, for any specified example, the probability of selection is $\pi_\eta = m/N$, and the probability that an original label y is presented to the student as label c is

$$U_\eta(c \mid y) = (1 - \pi_\eta)\mathbf{1}\{c = y\} + \frac{\pi_\eta}{K-1}\mathbf{1}\{c \neq y\}.$$

For each fixed y, these K nonnegative probabilities sum to one.

B.2 Decision-Rule Preservation

Let $p_y(x) = P(Y = y \mid X = x)$ denote the original class posterior. Uniform label noise leaves the input marginal unchanged and transforms the posterior to

$$q_{\eta,y}(x) = \left(1 - \frac{K\pi_\eta}{K-1}\right) p_y(x) + \frac{\pi_\eta}{K-1}.$$

For any two classes y and y', their posterior difference therefore satisfies

$$q_{\eta,y}(x) - q_{\eta,y'}(x) = \left(1 - \frac{K\pi_\eta}{K-1}\right)\big(p_y(x) - p_{y'}(x)\big).$$

Every posterior ordering, including ties, is preserved when $\pi_\eta < (K-1)/K$, so the set of Bayes decision rules is unchanged. At equality, every corrupted class probability is $1/K$; above equality, every strict posterior ordering is reversed. Sampling exactly $\lfloor \eta N \rfloor$ examples without replacement makes the selection events dependent, but each example still has marginal selection probability $\pi_\eta = \lfloor \eta N \rfloor / N$. The exact finite-dataset condition is consequently $\lfloor \eta N \rfloor / N < (K-1)/K$. For the ten-class MNIST and CIFAR-10 tasks, all reported corruption levels through 50% satisfy this condition.

B.3 Estimated Conditional-Entropy Coordinates

We use the calibrated out-of-fold teacher ensemble from Appendix A.2. "Out of fold" means that no teacher used to score an example was trained on that example. We average only over the student-training set after removing the validation split: $N = 55{,}000$ examples for MNIST and $N = 45{,}000$ for CIFAR-10. Each example has predictions from $S = 5$ ensemble members. Let

$$p_{i,s}(y) = \mathrm{softmax}(z_{i,s}/T^*)_y$$

be calibrated teacher member s's probability for class y on example i.

Applying the uniform relabeling rule to each member's original class probabilities gives

$$p^\eta_{i,s}(c) = (1 - \pi_\eta)p_{i,s}(c) + \frac{\pi_\eta}{K-1}\big(1 - p_{i,s}(c)\big).$$

Let $\Pi^\eta_{\mathcal{D}}$ denote the distribution of predictors obtained by applying this same relabeling rule to $G \sim \Pi_{\mathcal{D}}$ from Section 3. The transformed teacher ensemble approximates draws from $\Pi^\eta_{\mathcal{D}}$, so the mean member entropy estimates $A(x; \Pi^\eta_{\mathcal{D}})$ and serves as a model-based proxy for the pointwise Bayes-optimal risk under the corrupted-label distribution P_η. Because label corruption leaves the input

marginal unchanged, we estimate $\mathcal{R}^\star(P_\eta)$ by averaging these pointwise estimates over the student-training inputs. Following the uncertainty estimate in Appendix A.2, we average the entropy of the transformed probabilities from individual teachers rather than first averaging their probabilities:

$$\widehat{H}_\eta = \frac{1}{NS}\sum_{i=1}^{N}\sum_{s=1}^{S}\left[-\sum_{c=1}^{K} p_{i,s}^{\eta}(c)\log p_{i,s}^{\eta}(c)\right]. \tag{6}$$

We report the absolute estimate $\widehat{H}_\eta$, retaining the nonzero original-data baselines of 0.049416 nats for MNIST and 0.106292 nats for CIFAR-10. These are calibrated-ensemble estimates, not identified ground-truth conditional entropies: independently trained teachers are only an approximation to draws from a parameter posterior, and calibration or model misspecification can affect the numerical scale.

B.4 MNIST Student Sweep

We reserve a deterministic 5,000-example validation split from the 60,000 official MNIST training examples. The remaining 55,000 examples form the student-training set, and we evaluate on the clean 10,000-example official test set. The student is a two-hidden-layer ReLU MLP with topology

$$784 \to h \to h \to 10, \qquad h \in \{32, 64, 128, 256, 512, 1024, 2048, 4096\}.$$

Each width/noise condition uses three seeds, batch size 256, AdamW with weight decay 0.1, and cosine learning-rate decay.

Widths 32–4096 reuse the clean-data learning-rate choices from the historical uniform-label-noise sweep:

$$\{32 : 0.003,\ 64 : 0.003,\ 128 : 0.003,\ 256 : 0.003,$$
$$512 : 0.001,\ 1024 : 0.001,\ 2048 : 0.001,\ 4096 : 0.0003\}.$$

That pilot trained dense models for 200 epochs with three seeds but inherited AdamW weight decay 10^{-4}, so these rates were not reselected under weight decay 0.1. Each width's selected rate is fixed across all label-noise levels.

The epoch budget comes from a 300-epoch clean-data memorization pilot over the width grid. The largest average memorization time was 193 epochs at $h = 32$; the smallest multiple of 25 above it is 200, which we use for dense training. Each pruning round finetunes for half that duration, 100 epochs. We apply global iterative magnitude pruning with a minimum of 22 pruning operations, removing 20% of the remaining kernel weights by magnitude at each operation while leaving biases unpruned; the optimizer and cosine schedule restart for each finetuning phase. After this minimum horizon, pruning continues until the adaptive stopping rule in Appendix B.8 is satisfied. We save the dense checkpoint and every pruning checkpoint.

B.5 CIFAR-10 Student Sweep

We use a base-seed stratified split of the 50,000 official CIFAR-10 training examples. The student-training set contains 45,000 examples after removing 5,000 validation examples, and we evaluate on the clean 10,000-example official test set. The student family is WRN-28-k (Zagoruyko & Komodakis, 2016): the depth is fixed at 28 and the widen factor k scales the three residual-group channel counts to $(16k, 32k, 64k)$. A broad architecture-selection and memorization sweep considers

$$k \in \left\{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 10, 16\right\};$$

the label-noise experiment uses

$$k \in \{0.5, 1, 2, 4, 8\},$$

which spans widths below, at, and above the smallest clean-data memorizing widen factor, $k = 1$. Every condition uses three seeds, batch size 128, AdamW weight decay 0.1, five learning-rate warmup epochs, and cosine decay to 0.01 times the initial rate.

For each k, we select the initial learning rate on clean data from

$$\left\{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\right\}.$$

Each candidate is trained for 200 epochs with three seeds. Selection maximizes mean validation accuracy, breaking ties by lower validation loss and then lower learning rate. We then run a separate 1,000-epoch, three-seed clean-data memorization sweep over k.

For the label-noise runs, dense training stops when full unaugmented training accuracy first reaches exactly 100% or at the 1,000-epoch cap. We apply global iterative magnitude pruning, removing 20% of the remaining prunable parameters per round and finetuning for 60 epochs after each pruning operation. The minimum pruning horizon is 22 operations for $k \in \{0.5, 1, 2, 4\}$ and 32 operations for $k = 8$. All trajectories then follow the adaptive stopping rule in Appendix B.8.

Compute. Training and pruning runs execute on Google Cloud TPU v4-8, v5litepod-32, and v6e-4 accelerators.

B.6 Matching and Aggregation

For each architecture/noise cell, the test-accuracy matching threshold is the three-seed dense mean minus one population standard deviation. For each seed, we linearly interpolate test accuracy and remaining parameter count between adjacent checkpoints, including the dense-to-first-pruned segment. Pruned network size is the smallest qualifying measured count or interpolated threshold crossing on that trajectory; if neither exists, the zero-sparsity checkpoint is used as a fallback. The crossing is the continuous limit of the strict-greater-than criterion. Interpolation is linear in retained parameter count, equivalently in sparsity within a fixed architecture; no checkpoint weights are interpolated or evaluated, and there is no extrapolation. Plots report the mean count over the three seeds with a seed minimum–maximum band and compare uncertainty levels only within a fixed architecture size. Dense-capability markers require all three dense seeds in the cell to reach exactly 100% training accuracy.

B.7 Expected Test Accuracy Under Uniform Label Corruption

We corrupt only the training labels and evaluate model predictions against the original test labels. Expected test accuracy under the same uniform relabeling kernel U_η can be computed analytically from these recorded accuracies. For a fixed trained network, let $A^{\text{orig}} \in [0, 1]$ denote its accuracy on the original test labels and let A_η denote its accuracy after random test-label corruption. The expectation below holds the network and original test examples fixed and averages only over test-label corruption. An originally correct prediction remains correct with probability $1 - \pi_\eta$, while an originally incorrect prediction becomes correct with probability $\pi_\eta/(K-1)$. Linearity of expectation therefore gives

$$\begin{aligned} \mathbb{E}[A_\eta] &= (1-\pi_\eta)A^{\text{orig}} + \frac{\pi_\eta}{K-1}\big(1 - A^{\text{orig}}\big) \\ &= \alpha_\eta A^{\text{orig}} + \beta_\eta, \qquad \alpha_\eta = 1 - \frac{K\pi_\eta}{K-1}, \quad \beta_\eta = \frac{\pi_\eta}{K-1}. \end{aligned} \tag{7}$$

This calculation requires only the recorded original-label accuracy, so no additional model inference is needed. It does not require independent corruption events across examples. If exactly $\lfloor \eta M \rfloor$ labels are corrupted in a test set of size M, the same formula uses the marginal probability $\lfloor \eta M \rfloor / M$ in place of π_η; both probabilities equal η for our evaluated noise grid and dataset sizes.

For $K = 10$ and our evaluated corruption fractions through 50%, $\alpha_\eta > 0$. Fix an architecture and training corruption level, and let μ and σ be the mean and population standard deviation of original-label test accuracy across the three zero-sparsity replicas. The original reference threshold is $\tau^{\text{orig}} = \mu - \sigma$. Computing the reference threshold from the replicas' expected corrupted-label accuracies gives

$$\tau_\eta^{\text{exp}} = (\alpha_\eta \mu + \beta_\eta) - \alpha_\eta \sigma = \alpha_\eta \tau^{\text{orig}} + \beta_\eta.$$

The positive affine transformation therefore preserves every strict threshold comparison and equality. It also preserves interpolated threshold crossings: for adjacent original-label checkpoint accuracies $a \neq b$, the crossing fraction satisfies

$$\frac{\tau_\eta^{\text{exp}} - (\alpha_\eta a + \beta_\eta)}{\alpha_\eta (b - a)} = \frac{\tau^{\text{orig}} - a}{b - a}.$$

The checkpoint parameter counts remain fixed, so the qualifying measured counts, interpolated crossing counts, and zero-sparsity fallbacks are unchanged. Consequently, computing both checkpoint scores and reference thresholds from expected corrupted-label accuracies yields the same per-seed pruned network sizes, and hence the same plotted means and seed ranges, as evaluation against the original test labels. This equivalence concerns the condition-specific matching criterion in Appendix B.6 applied to expected accuracies. It does not assert that a particular sampled corrupted test set yields the same pruned network size, or that averaging selected sizes over such test sets gives the same result.

B.8 Adaptive Pruning Horizon

The fixed-accuracy overhead and condition-specific pruned-network-size analyses use the same adaptively terminated uniform-label-noise trajectories. The latter includes every recorded checkpoint for each seed, even when some seeds continue beyond the last round recorded for all three seeds. Every zero-sparsity and fully finetuned positive-sparsity checkpoint is evaluated on the unchanged 10,000-example official test set of its dataset. We retain eight MNIST widths $h \in \{32, 64, 128, 256, 512, 1024, 2048, 4096\}$ and all five CIFAR-10 widen factors $k \in \{0.5, 1, 2, 4, 8\}$, with nine noise levels and three seeds per architecture/noise cell.

For each architecture a, let $A_{a,\eta,\xi,r}$ be original-test-set accuracy for training label-noise fraction η, run ξ, and pruning round r, where $r = 0$ denotes the zero-sparsity checkpoint. Let Ξ denote the set of three run identifiers, paired by seed across corruption conditions. We define the stopping target as

$$T_a = \frac{1}{3} \sum_{\xi \in \Xi} A_{a,0.5,\xi,0}.$$

MNIST trajectories and CIFAR-10 trajectories with $k \in \{0.5, 1, 2, 4\}$ use a minimum horizon of 22 pruning operations, while CIFAR-10 trajectories with $k = 8$ use a minimum horizon of 32 operations. After the minimum horizon, each seed stops once three consecutive fully finetuned checkpoints have accuracy strictly below T_a; equality or a rebound to or above T_a resets the consecutive-below-target count. The safety limit is 100 total pruning rounds; no retained trajectory reaches that limit, and all retained trajectories satisfy the stopping rule. The stopping rule uses test accuracy to determine how far to measure each trajectory; it does not introduce a validation-based model-selection criterion or replace the original mean-minus-standard-deviation matching rule.

WRN-28-8 Minimum Horizon. The longer minimum horizon prevents the widest CIFAR-10 model from stopping in an intermediate accuracy valley. At 30% label corruption, its original 22-operation trajectory ends at approximately 64.9% mean test accuracy, below its approximately 76.4% zero-sparsity reference, but further pruning recovers to approximately 81.0% before operation 32. A common fixed 22-operation horizon would therefore miss the subsequent recovery and contraction. The adaptive trajectories in Figure 14 retain this recovery and the later decline.

Let $M_{a,\eta,\xi,r}$ denote remaining parameter count. In the appendix trajectory views, at rounds with all three seeds recorded, solid curves plot the three-seed mean accuracy against the three-seed mean parameter count, and bands span the seed accuracy minimum and maximum. Each plotted curve ends at the last round with all three seeds recorded; subsequent individual-seed checkpoints are omitted, and no reduced-seed mean is shown. The trajectory figures include the zero-sparsity checkpoints and all pruning rounds with complete three-seed support. Tables 2 and 3 record the final measured rounds for all three seeds, including those beyond the plotted curves.

Colors encode the estimated post-corruption conditional entropy $\widehat{H}_\eta$ from Equation (6), rather than the corrupted-label fraction or added entropy. The common linear color scale spans 0–2 nats for both datasets. Lines connect consecutive recorded checkpoints without smoothing, extrapolation, or an accuracy envelope.

B.9 Fixed-Accuracy Uncertainty Overhead

For each architecture, restrict to pruned rounds $r > 0$ with all three runs recorded and retain the run-specific accuracies and parameter counts $A_{\eta,\xi,r}$ and $M_{\eta,\xi,r}$. For target accuracy t, linearly interpolate every adjacent pair $(A_{\eta,\xi,r}, M_{\eta,\xi,r})$ and define $\widetilde{M}_{\eta,\xi}(t)$ as the smallest parame-

ter count among qualifying measured checkpoints and segment crossings. We pair runs with the same seed across each aleatoric uncertainty condition and the zero-corruption baseline and compute the signed parameter difference, $\Delta_{\eta,\xi}(t) = \widetilde{M}_{\eta,\xi}(t) - \widetilde{M}_{0,\xi}(t)$. The plotted line is the mean of $\Delta_{\eta,\xi}(t)$ over $\xi \in \Xi$, and the shaded region spans the observed minimum and maximum across those three runs. The baseline has zero label corruption, rather than zero estimated intrinsic conditional entropy. The full-range appendix views share horizontal limits covering the union of the architecture-specific accuracy domains; the 75%-and-above views retain the same upper limit. A condition is shown only through the greatest target supported by every condition trajectory and its paired zero-corruption trajectory. A checkpoint remains eligible at equality. Unlike the condition-specific matching calculation, this fixed-accuracy diagnostic excludes zero-sparsity checkpoints and rounds without complete three-seed support, and it uses no fallback. Neither calculation extrapolates beyond the measured trajectories or evaluates interpolated network weights. The shaded region is the observed three-seed range, not a confidence interval. The accuracy axis is linear; the signed parameter-count axis is base-10 symmetric-log, with a linear interval of $[-100, 100]$ parameters for MNIST and $[-1000, 1000]$ parameters for CIFAR-10. The reproducible renderer is `nsc/sparse/analysis/reporting/generate_interpolated_signed_uncertainty_overhead.py`; its figure provenance pages record the input hash and invocation.

C Uniform Label-Noise Results

This appendix presents the complete architecture-grid evidence for uncertainty overhead, target relaxation, and the two regimes, following the order of Sections 5, 6, and 7. Appendix D.3 examines sensitivity to the checkpoint-selection criterion.

C.1 Uncertainty Overhead Across Architecture Sizes

Figures 6 and 7 extend the main fixed-accuracy comparison to every retained architecture over the complete supported accuracy range. Each curve fixes a training aleatoric uncertainty level and reports the signed difference in pruned network size from the zero-corruption condition at the same original-test-set accuracy. Figures 8 and 9 regroup the same measurements by aleatoric uncertainty above 75% accuracy, making the dependence on architecture size easier to compare.

Curves are paired three-seed means, and shaded bands span the observed seed range. Each condition stops at the greatest accuracy target supported by all three paired seeds, without extrapolation. The fixed-accuracy calculation uses positive-sparsity checkpoints from rounds with complete three-seed support, as detailed in Appendix B. The horizontal limits are shared within each figure, and the symmetric-log vertical axes have linear regions of $[-100, 100]$ parameters for MNIST and $[-1000, 1000]$ parameters for CIFAR-10.

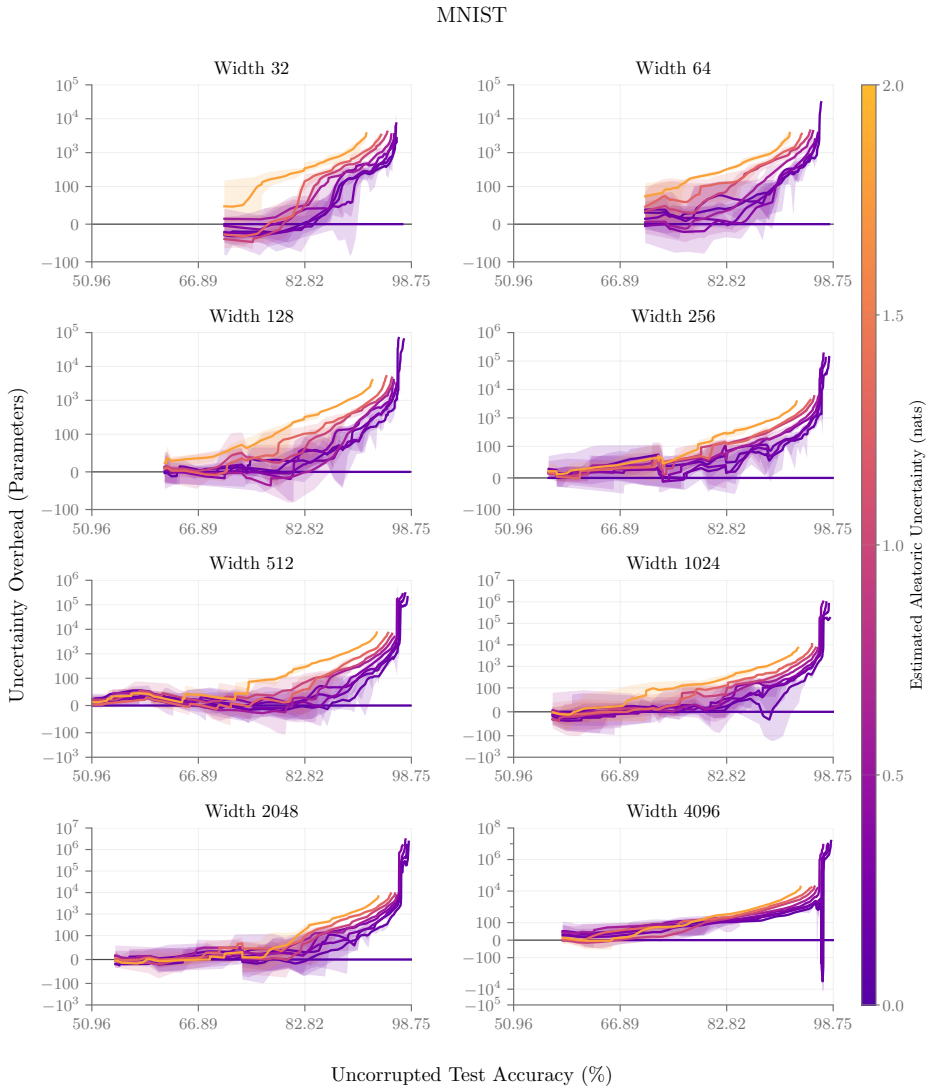

Figure 6: Signed parameter-count uncertainty overhead for all eight retained MNIST widths on the shared 50.96–98.7533% complete uncorrupted-test-accuracy axis.

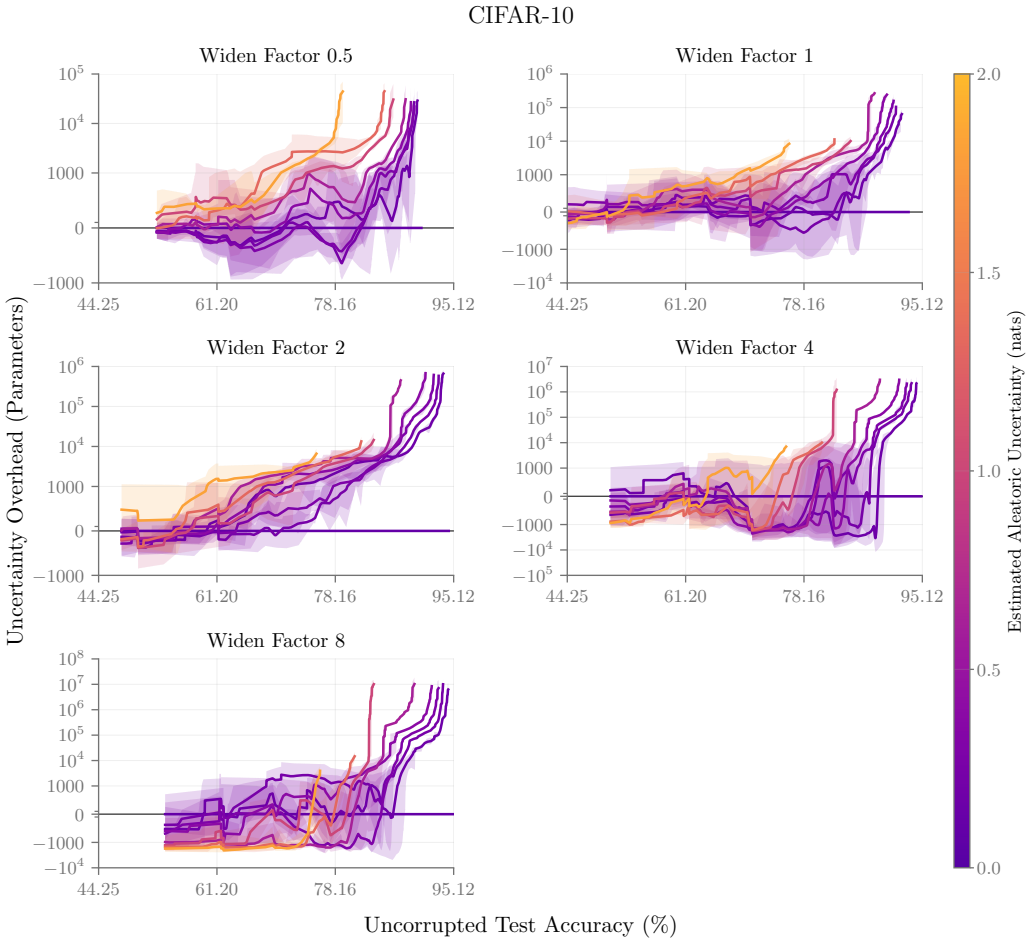

Figure 7: Signed parameter-count uncertainty overhead for all five retained CIFAR-10 widen factors on the shared 44.2467–95.12% complete uncorrupted-test-accuracy axis.

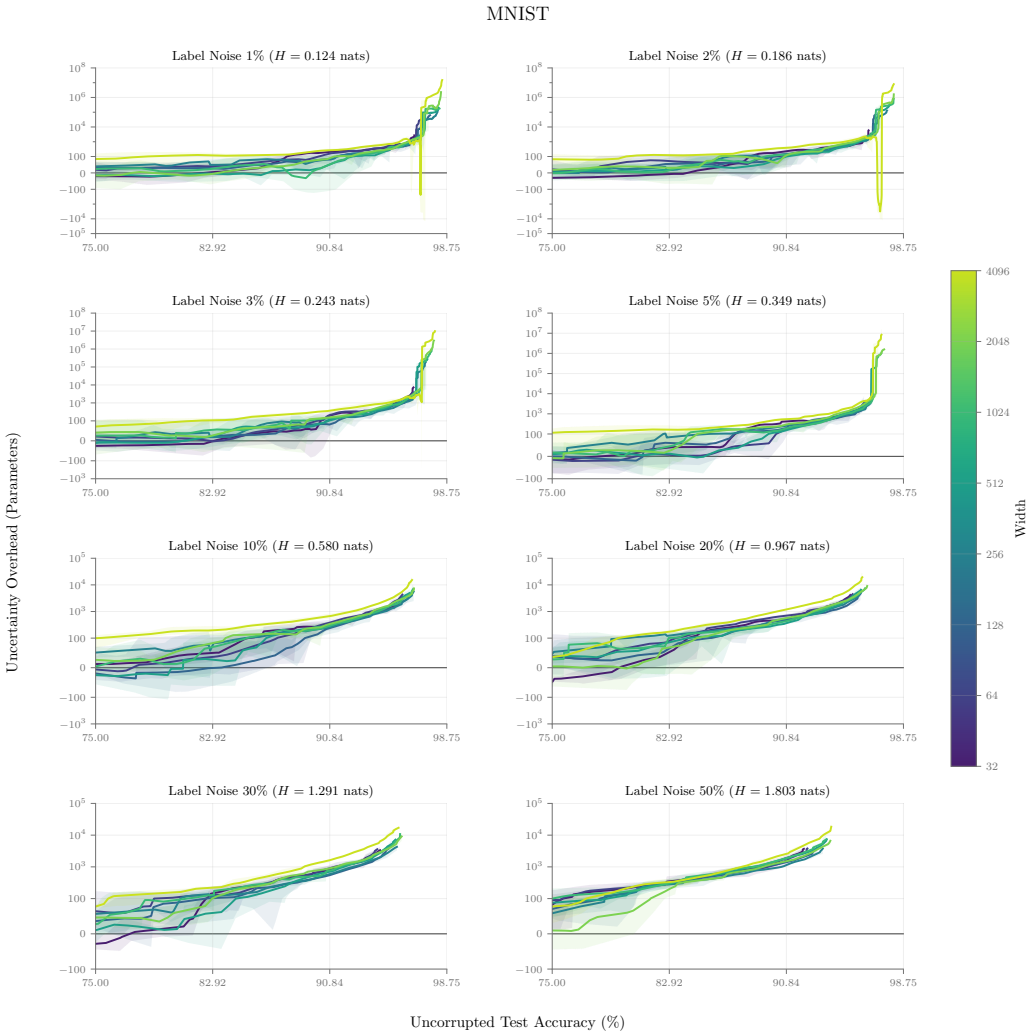

Figure 8: MNIST signed uncertainty overhead in parameters, grouped by estimated aleatoric uncertainty for all eight retained widths from 75% uncorrupted test accuracy upward. Every panel shares the 75–98.7533% accuracy interval.

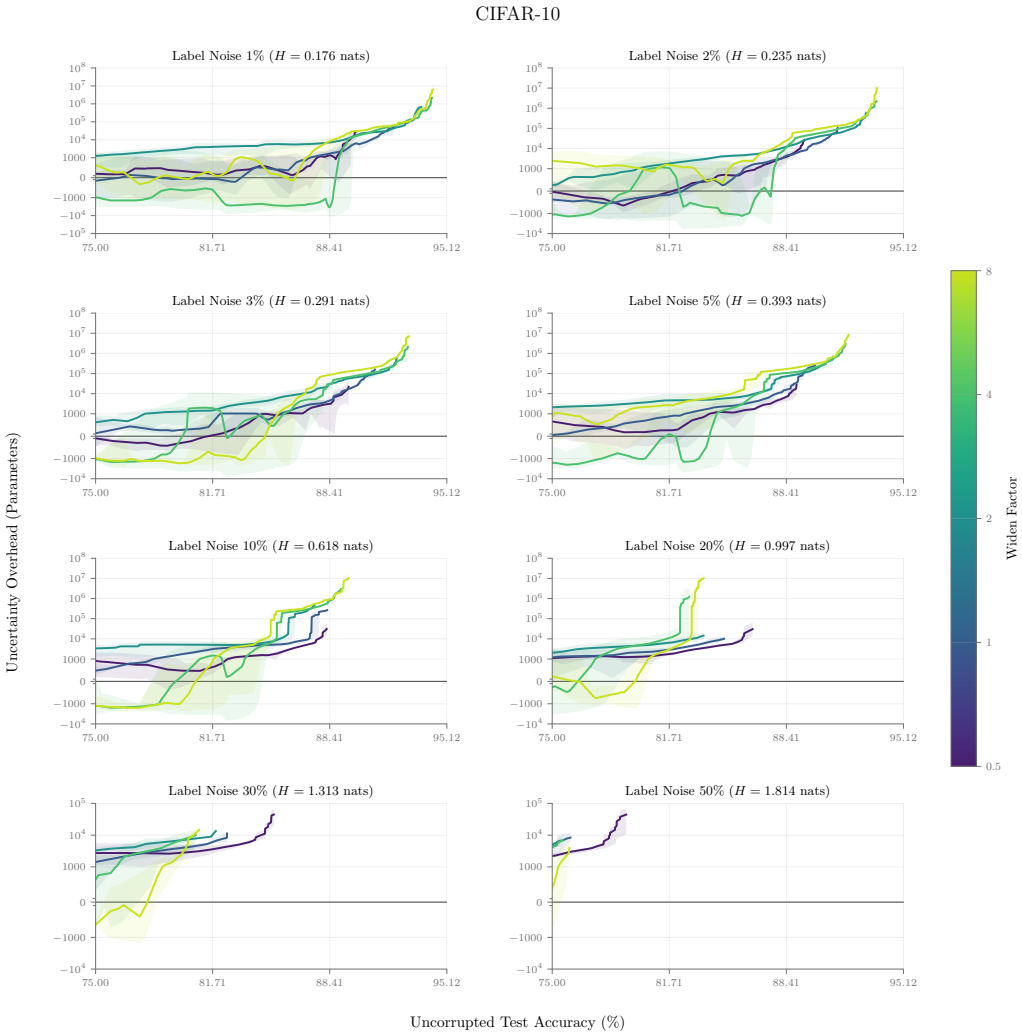

Figure 9: CIFAR-10 signed uncertainty overhead in parameters, grouped by estimated aleatoric uncertainty for all five retained widen factors from 75% uncorrupted test accuracy upward. Every panel shares the 75–95.12% accuracy interval.

C.2 Target Relaxation Across Architecture Sizes

Figure 5 in the main text shows the dense accuracies underlying the target changes in Figure 3. The dense-accuracy curves report three-seed means with minimum–maximum ranges, while each matching target is the corresponding mean minus one population standard deviation. Figures 10 and 11 reorient these accuracies by architecture size at each aleatoric uncertainty level. They show why the dependence of target relaxation on architecture size is not monotone: at high aleatoric uncertainty, mean dense accuracy initially falls with architecture size and then recovers among the largest architectures.

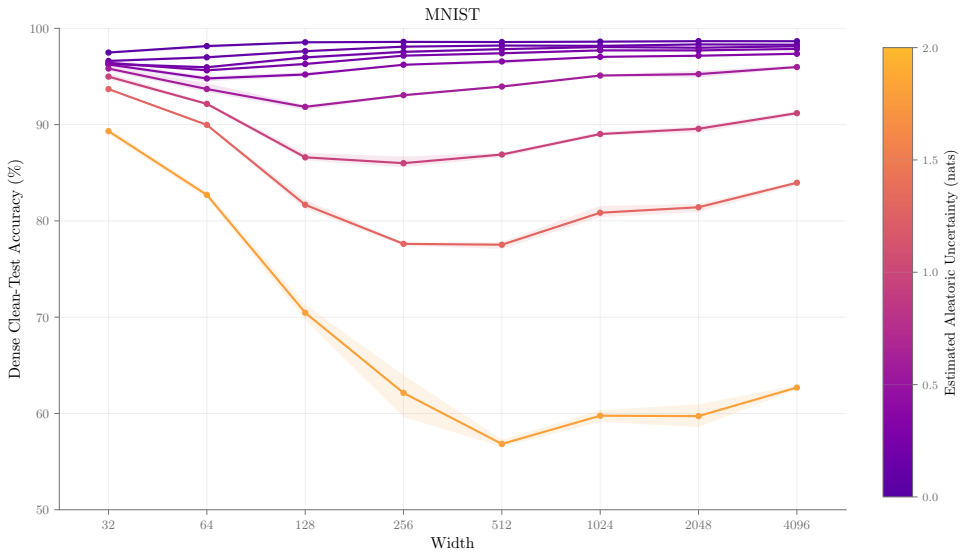

Figure 10: MNIST zero-sparsity clean-test accuracy against hidden width for all nine uniform-label-noise conditions. Curves are colored by calibrated estimated aleatoric uncertainty on the shared 0–2 nat scale. Points and connecting lines show three-seed means, shaded bands span observed seed minima and maxima, and hidden width is on a base-two logarithmic axis.

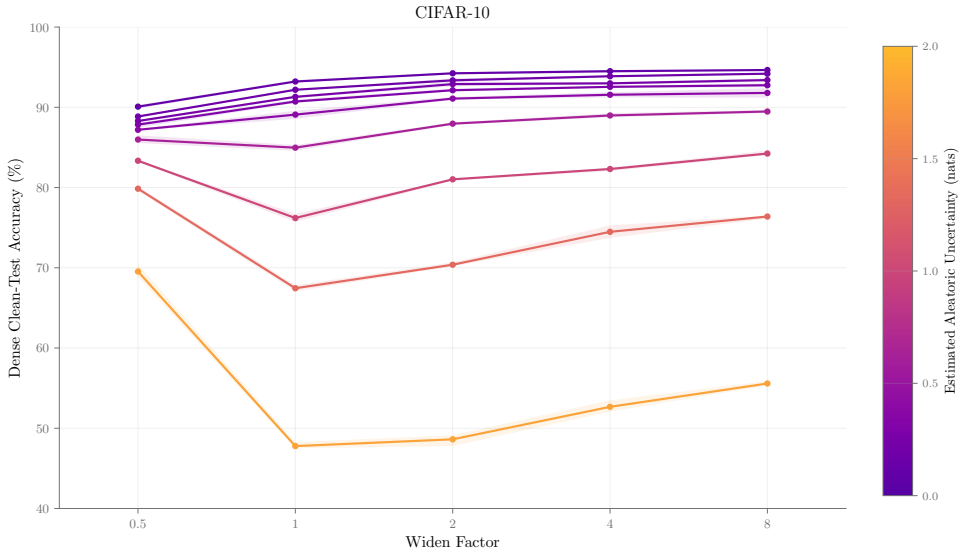

Figure 11: CIFAR-10 zero-sparsity clean-test accuracy against WRN-28 widen factor for all nine uniform-label-noise conditions. Curves, colors, three-seed aggregation, and the base-two logarithmic architecture axis follow Figure 10.

C.3 Two-Regime Results and Pruning-Trajectory Diagnostics

Figure 12 reorients the main pruned-network-size results to show their dependence on architecture size at fixed aleatoric uncertainty. For MNIST width 4096, mean pruned network size rises from 91,355 parameters without label corruption to 8,079,851 at 5% corruption, then falls to 8,359 at 50%. For CIFAR-10, WRN-28-0.5 contracts throughout the nonzero-corruption range, from 30,947 parameters without label corruption to 3,763 at 50%. WRN-28-8 peaks at 20% corruption and falls to 23,667 and 16,058 parameters at 30% and 50%, respectively, below its original fixed-horizon floor of 32,868 parameters.

All 216 MNIST seed trajectories and 134 of 135 CIFAR-10 seed trajectories produce a qualifying pruned-network-size selection without fallback. The remaining trajectory uses the zero-sparsity fallback at CIFAR-10 widen factor 2, 10% corruption, seed 0. These selections include measured checkpoints and interpolated threshold crossings; they do not assert the measured performance of any unobserved network.

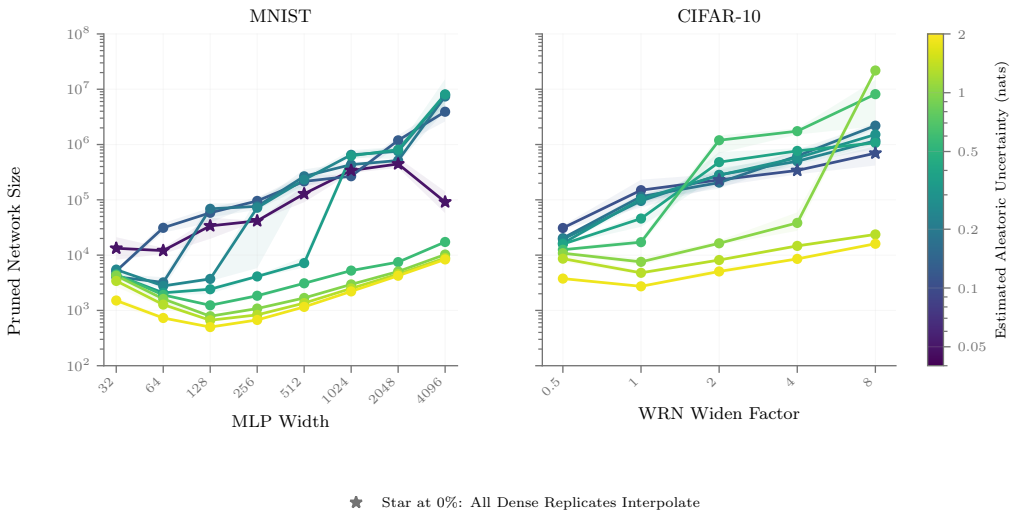

Figure 12: Pruned network size as a function of architecture size at fixed estimated aleatoric uncertainty for MNIST (left) and CIFAR-10 (right) under uniform label noise. This is the transposed view of Figure 4. Both axes are logarithmic; the horizontal axes show MLP width and WRN-28 widen factor, respectively. Each curve fixes a label-noise level, and its color encodes the corresponding estimated aleatoric uncertainty in nats. Points are three-seed means, and shaded bands span the minimum and maximum across seeds. Stars at 0% label noise mark architectures for which all dense replicates interpolate the training data.

C.3.1 Pruning Trajectories and Observation Depth

Figures 13 and 14 show the complete architecture grids of pruning trajectories used for the fixed-accuracy and condition-specific analyses. Each plotted curve includes zero-sparsity and positive-sparsity checkpoints and ends at the last round recorded for all three seeds. The condition-specific pruned-network-size calculation additionally includes later checkpoints recorded for individual seeds. Tables 2 and 3 give the final measured rounds for every architecture and label-noise fraction, including these individual-seed tails. Each cell lists seed offsets $0/1/2$ from base seed 1337; the entries count completed pruning operations and exclude the zero-sparsity checkpoint. Every retained trajectory satisfies the adaptive stopping rule in Appendix B.8.

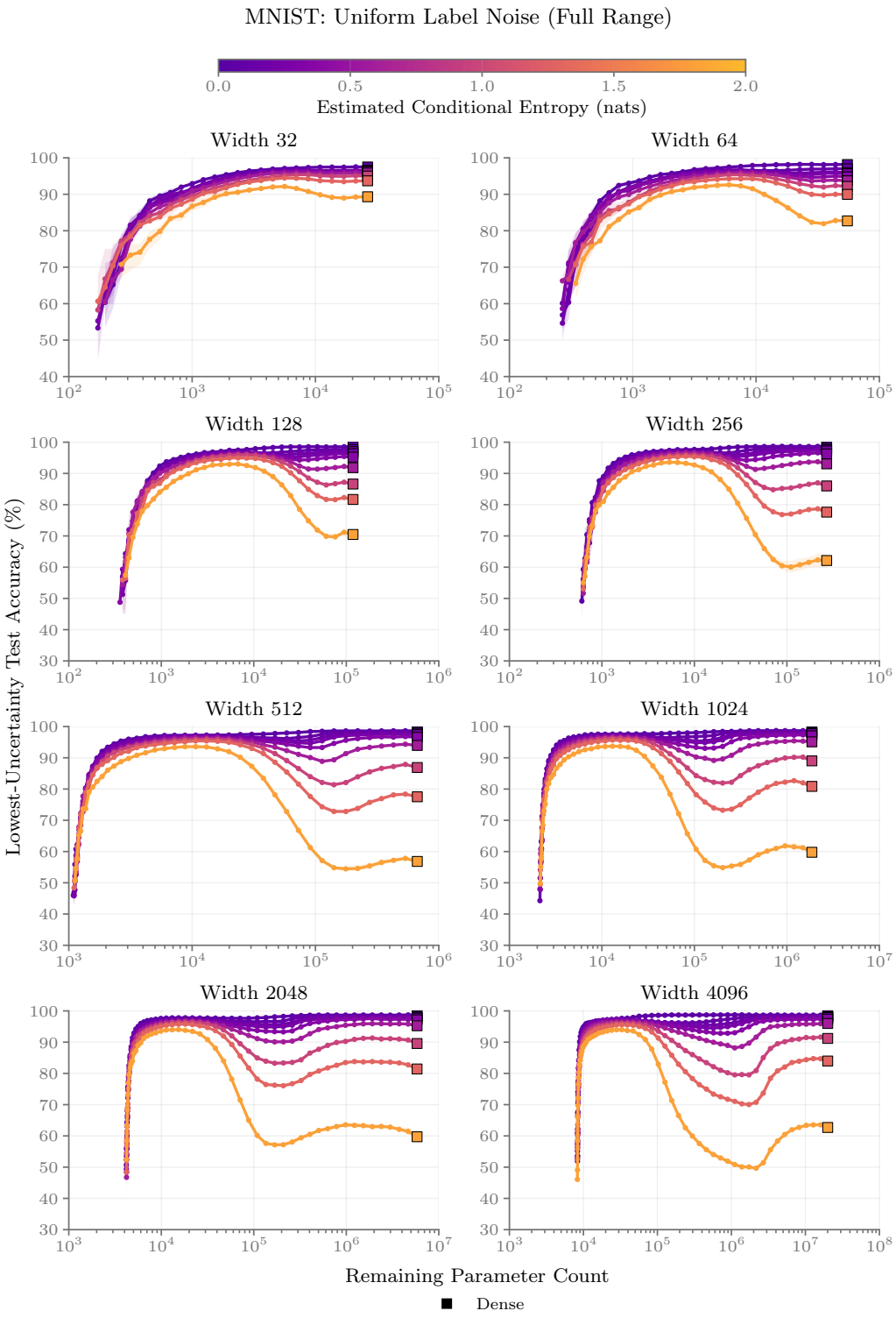

Figure 13: MNIST pruning trajectories for all eight retained hidden widths. Squares mark zero-sparsity checkpoints; curves show three-seed mean test accuracy against mean retained parameter count, and bands span the observed seed accuracy range. Colors encode estimated aleatoric uncertainty on a shared 0–2 nat scale. Only rounds with complete three-seed support are shown, and the parameter-count axis is logarithmic.

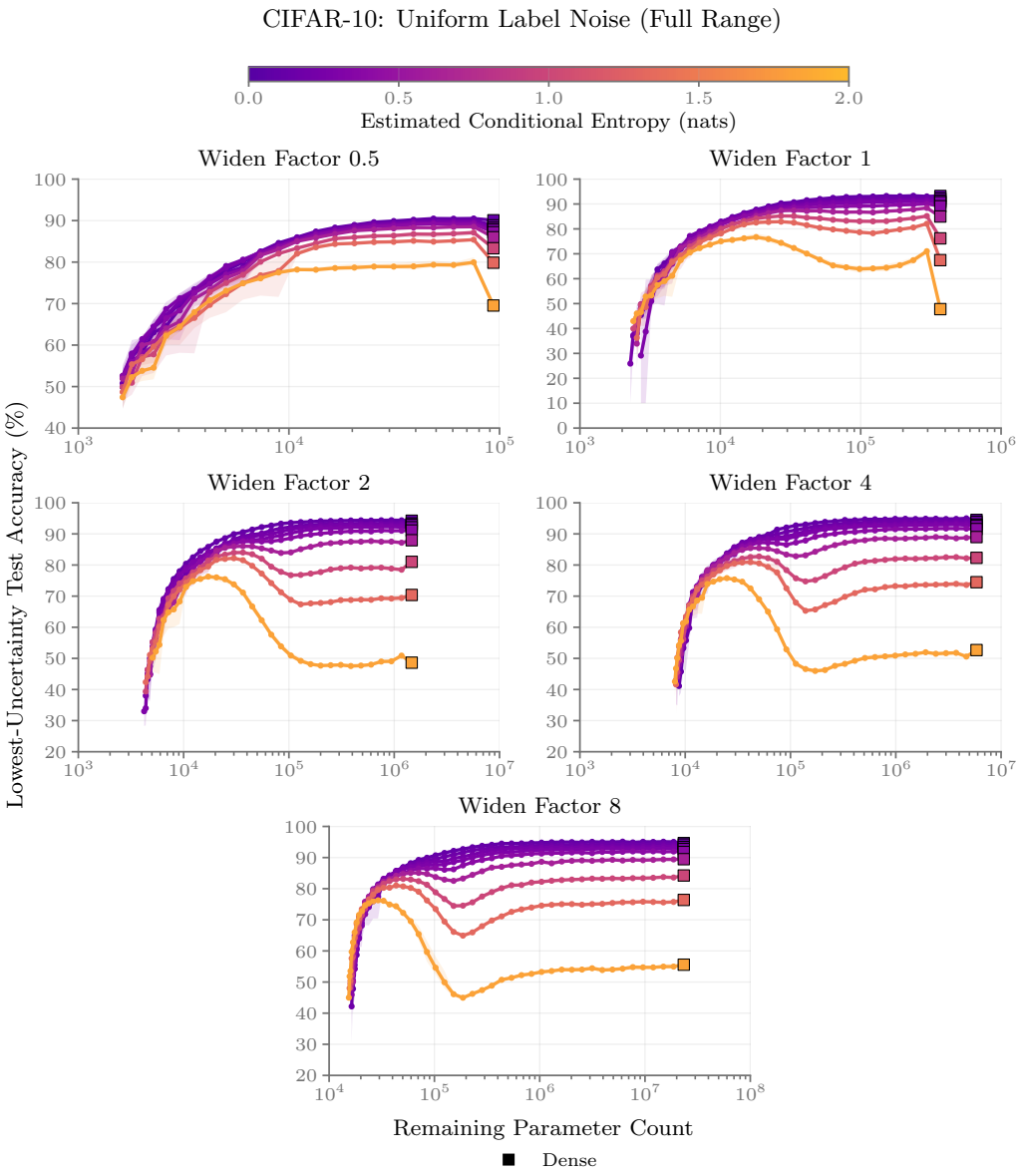

Figure 14: CIFAR-10 pruning trajectories for all five retained WRN-28 widen factors. Checkpoint markers, aleatoric uncertainty colors, and three-seed means and ranges follow Figure 13. Only rounds with complete three-seed support are shown.

Table 2: Final total pruning rounds for MNIST by hidden width and training label-noise fraction. Entries give seed offsets $0/1/2$ separately.

Width	0%	1%	2%	3%	5%	10%	20%	30%	50%
32	25/24/24	24/25/25	25/25/25	25/25/25	24/26/24	24/24/24	27/25/25	25/25/25	22/23/24
64	27/28/27	27/27/28	28/27/27	26/27/27	27/28/29	28/27/28	28/26/27	27/25/27	25/26/27
128	31/32/31	31/31/31	31/32/32	32/30/31	32/32/32	31/32/31	31/30/31	32/32/31	30/31/30
256	36/38/36	35/37/36	37/37/35	36/35/38	37/36/35	35/37/36	35/37/37	35/37/35	36/36/35
512	43/43/42	41/42/41	41/40/40	41/41/42	40/42/42	40/40/41	40/41/42	42/41/41	42/40/41
1024	44/43/46	46/45/45	46/45/43	46/46/45	44/46/45	46/46/43	44/47/44	46/44/44	44/45/45
2048	50/49/50	51/50/51	49/50/50	49/50/51	51/50/50	49/49/51	49/51/51	50/49/49	50/49/50
4096	55/55/54	54/55/54	54/54/54	55/55/54	53/54/56	53/53/56	56/55/56	54/55/54	55/55/56

Table 3: Final total pruning rounds for CIFAR-10 by WRN-28 widen factor and training label-noise fraction. Entries follow Table 2. Widen factor 0.5 stops at its minimum horizon for every seed and noise condition.

Widen Factor	0%	1%	2%	3%	5%	10%	20%	30%	50%
0.5	22/22/22	22/22/22	22/22/22	22/22/22	22/22/22	22/22/22	22/22/22	22/22/22	22/22/22
1	29/30/28	31/29/30	30/30/30	29/27/27	29/30/28	30/30/28	30/29/30	30/28/30	29/29/29
2	33/34/35	35/33/34	34/36/34	36/35/35	34/33/34	34/34/33	35/34/36	35/35/34	35/31/36
4	38/38/37	36/38/39	39/39/39	39/38/39	37/38/40	40/38/40	40/39/38	39/41/40	40/40/40
8	42/43/41	42/43/44	41/43/44	43/43/42	43/42/44	45/43/43	44/44/43	45/44/45	45/45/45

D Robustness and Alternative Explanations

We examine sensitivity to initial model size, the rule used to remove weights, and the criterion used to select a checkpoint from a pruning trajectory. These comparisons distinguish the reported pruned network sizes from intrinsic lower bounds on the capacity required by a task.

D.1 Sensitivity to Initial Model Size

As aleatoric uncertainty increases, a fixed architecture can move closer to its condition-specific interpolation threshold. One might therefore wonder whether uncertainty overhead merely reflects this reduction in relative overparameterization: perhaps a model closer to the interpolation threshold has a larger pruned network size even if aleatoric uncertainty is otherwise irrelevant.

We test this alternative explanation by varying initial model size while holding both the data distribution and the dense-reference threshold fixed. Within each dataset and label-noise condition, every initial architecture supplies one dense-reference threshold equal to its three-seed mean dense test accuracy minus one population standard deviation. We apply each dense-reference threshold to the pruning trajectories of every initial architecture. The resulting family of curves spans the dense test accuracies attained by models of different sizes without changing the aleatoric uncertainty condition. Moving leftward along a curve reduces initial model size and serves as a controlled proxy for moving closer to the interpolation threshold.

Figure 15 gives the comparison with no added label noise. Across the spectrum of dense-reference thresholds, reducing initial model size generally reduces pruned network size. The relationship is not perfectly monotone—most visibly around some intermediate MNIST widths—but the dominant direction is opposite the proposed confound: moving toward the interpolation threshold generally decreases, rather than increases, pruned network size.

We repeat the same controlled comparison independently at every evaluated label-noise level; Figure 16 shows the 10% condition. Restricting the summary to adjacent initial-size comparisons for which all three model seeds meet the dense-reference threshold and neither endpoint is floor-

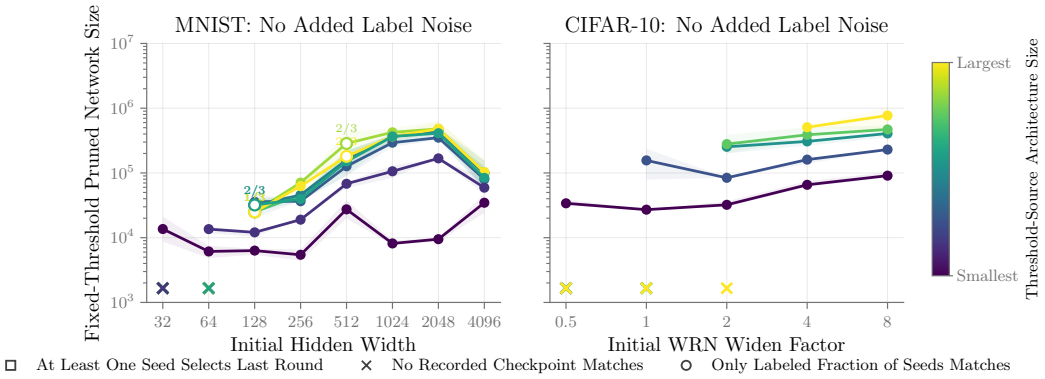

Figure 15: Initial-model-size sensitivity with no added label noise. Each curve fixes a dense-reference threshold supplied by one dense architecture, and color orders the threshold-source architectures from smallest to largest. Lines are means over matched model seeds, and bands span their minimum and maximum pruned network sizes. Hollow markers identify an initial architecture for which fewer than three model seeds meet the dense-reference threshold, crosses identify an initial architecture for which no checkpoint meets the dense-reference threshold, and square outlines identify a floor-censored selection.

censored, the larger initial architecture has the larger pruned network size in $311/374 = 83.2\%$ of MNIST comparisons and $115/120 = 95.8\%$ of CIFAR-10 comparisons. Table 4 shows that the same majority direction holds at every evaluated label-noise level in both datasets.

Table 4: Adjacent initial-size comparisons in which the larger initial architecture has the larger mean pruned network size. Denominators include only comparisons for which both architectures meet the fixed dense-reference threshold for all three model seeds and neither architecture is floor-censored.

Label Noise	MNIST	CIFAR-10
0%	23/34 (67.6%)	10/12 (83.3%)
1%	26/32 (81.2%)	10/11 (90.9%)
2%	26/32 (81.2%)	11/12 (91.7%)
3%	24/34 (70.6%)	11/11 (100.0%)
5%	26/33 (78.8%)	12/12 (100.0%)
10%	43/54 (79.6%)	11/11 (100.0%)
20%	49/56 (87.5%)	14/15 (93.3%)
30%	52/56 (92.9%)	20/20 (100.0%)
50%	42/43 (97.7%)	16/16 (100.0%)
All conditions	311/374 (83.2%)	115/120 (95.8%)

These measurements do not imply that pruned network size is universally monotone in initial model size. They instead test the direction required by the proposed alternative explanation. If moving closer to the interpolation threshold were itself responsible for uncertainty overhead, reducing initial model size would need to systematically increase pruned network size. We generally observe the reverse relationship across datasets, aleatoric uncertainty levels, and dense-reference thresholds. This control therefore argues against that explanation of uncertainty overhead, although changing initial architecture size is only a proxy for the effect of changing the interpolation threshold through label noise.

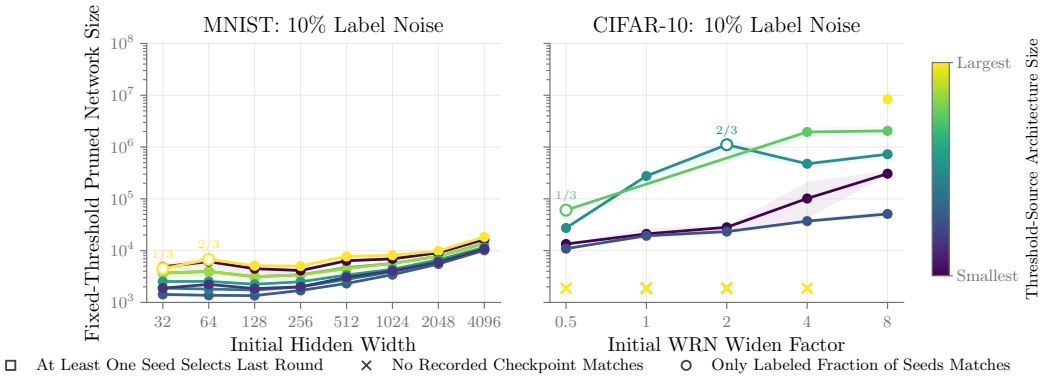

Figure 16: Initial-model-size sensitivity at 10% label noise. Encodings follow Figure 15.

D.2 Clean-Label Taylor Pruning Under Label Noise

We compare magnitude pruning with a clean-label Taylor pruning oracle to test whether the measured pruned network size under noisy supervision depends on the rule used to select weights for removal. The oracle has access to correct labels for a held-out validation set, so this comparison probes pruning-rule dependence rather than a deployable noise-robust pruning method.

Paired Experimental Design. We split the standard MNIST training set into a 55,000-example training pool and a 5,000-example clean-label validation set. We apply uniform forced-different label corruption to 0%, 3%, or 10% of the training-pool labels, leaving the validation and standard test labels unchanged. For each corruption level and three model seeds, we train a $784 \to 8192 \to 8192 \to 10$ MLP for 200 epochs and then run 22 rounds of global iterative pruning. Dense training and pruning-round finetuning use batches of 256, AdamW with weight decay 0.1, an initial learning rate of 3×10^{-4}, and cosine learning-rate decay. Each round removes 20% of the remaining kernel weights, leaves biases unpruned, and finetunes for 100 epochs on the same noisy training pool. The paired pruning rules use the same data split, realized corruption, initialization, minibatch order, optimizer settings, and finetuning schedule; only the weight-selection rule differs.

Magnitude pruning removes the active weights with the smallest absolute values $|w_j|$. Clean-label Taylor pruning instead removes the active weights with the smallest scores $|w_j \, \partial L_{\text{clean val}}/\partial w_j|$, where $L_{\text{clean val}}$ is mean cross-entropy on the held-out clean-label validation set. This score is the magnitude of a first-order estimate of the validation-loss change from setting w_j to zero before finetuning. The clean-label set determines only which weights are removed; both rules use the corrupted training labels for dense training and for every finetuning round.

Selection and Results. At each corruption level, we set one shared clean-test accuracy threshold to the paired dense models' mean clean-test accuracy minus 0.5 percentage points. For each trajectory, the pruned network size is the parameter count of the sparsest measured checkpoint meeting that threshold, including the zero-sparsity checkpoint as a candidate. Table 5 reports the three-seed means.

Table 5: Mean clean-test matching threshold and pruned network size over three paired model seeds at each label-corruption level. The reduction is the mean of the paired seedwise percentage reductions in parameter count, rather than the reduction computed from the displayed mean counts.

Label corruption	Threshold	Magnitude	Clean-label Taylor	Reduction
0%	98.137%	559,563	559,562	0.000%
3%	97.407%	4,063,329	559,562	86.229%
10%	95.293%	10,061,279	1,459,685	86.420%

At zero corruption, both rules reach the last measured pruning round for all three seeds, so this comparison does not resolve whether either rule could prune further in that condition. At 3% corruption, magnitude pruning selects round 12 and clean-label Taylor pruning selects round 21 for every seed. At 10% corruption, the selected rounds are 7–9 for magnitude pruning and 14–20 for clean-label Taylor pruning across the three seeds. At the rounds selected by magnitude pruning, clean-label Taylor pruning improves mean clean-test accuracy by 0.337 and 0.570 percentage points at 3% and 10% corruption, respectively, compared with 0.013 points at zero corruption.

This diagnostic establishes that the pruning rule can substantially change the measured pruned network size under label corruption, even when dense training and the subsequent optimization schedule are held fixed. It does not show that magnitude pruning specifically preserves weights used to fit corrupted labels: observed-label training accuracy at the same pruning rounds is equal between rules in all nonzero-corruption seed pairs. The comparison covers one MLP width, three corruption levels, three seeds, and 22 discrete pruning rounds, and the clean-label Taylor rule uses information that may be unavailable for naturally noisy data.

D.3 Sensitivity to Test-Optimal Checkpoint Selection

The primary criterion selects the smallest retained parameter count that reaches the condition-specific dense-reference accuracy target. As an alternative, we select the measured checkpoint with the highest test accuracy over each complete pruning trajectory, including the zero-sparsity checkpoint. Exact accuracy ties select the smallest parameter count, followed by the later pruning operation. We aggregate the selected counts across the three seeds for each architecture and corruption level.

Under this test-optimal criterion, both expansion and contraction remain present for 7 of 8 MNIST widths and 4 of 5 CIFAR-10 widen factors (Table 6). No selected checkpoint occurs at the adaptive pruning horizon, so the observed extrema are not selections at the final measured checkpoint. These results support the natural-label-noise transition under a second selection criterion, while leaving one architecture in each dataset without both regimes.

Table 6: Sensitivity of the uniform-label-noise results to test-optimal checkpoint selection. Regime counts compare mean selected parameter counts across corruption levels for each architecture; final-checkpoint counts include all individual seed trajectories.

Dataset	Architectures With Both Regimes	Selections at Final Checkpoint
MNIST	$7/8$	$0/216$
CIFAR-10	$4/5$	$0/135$

E Natural-Data Uncertainty-Reduction Methodology (Extended)

This appendix provides the complete methodology summarized in Section 8.

The calibrated ensemble protocol shared with the label-noise experiments is given in Appendix A.2.

E.1 Generated-Dataset Construction

For each dataset, let $r_{c,b}$ be the empirical probability mass of real training examples from class c in expected-entropy bin b, and let h_b be the bin center. For a scalar λ shared across all ten classes, we define

$$q_{\lambda,c,b} = \frac{r_{c,b} \exp(\lambda h_b)}{\sum_{b'} r_{c,b'} \exp(\lambda h_{b'})}. \tag{8}$$

We fix each class probability at $1/10$ and choose the shared λ to achieve the aggregate bin-center mean μ. At $\lambda = 0$, this construction recovers the empirical class-conditional histograms at the 0.025-nat bin resolution; negative values shift mass toward lower-entropy bins. Appendix E.7 shows

that this exponential tilt is the minimum-KL change to the reference histogram subject to a target mean. For both MNIST and CIFAR-10, only generated images whose ensemble prediction agrees with the class-conditioning label are eligible, and the hard conditioning label is used for training and evaluation.

Histogram discretization and materialization. We use 93 bins with edges $0, 0.025, \ldots, 2.325$ nats, covering the full ten-class entropy range $[0, \log 10]$. Bins are left-closed and right-open over this entropy range, and h_b denotes the midpoint of bin b. Reference counts are normalized separately within each class before tilting; bins with zero reference probability retain zero target probability. We convert the tilted probabilities into integer counts separately for the development and test sets using largest-remainder apportionment, with $5{,}000$ development and $1{,}000$ test examples per class. Within each class, we split the selected development examples into $4{,}500$ training and 500 validation examples. For each of six targets, we construct three dataset replicas, giving 18 family datasets per source dataset. Every replica contains $45{,}000$ training, $5{,}000$ validation, and $10{,}000$ matched-test images, with exact class balance in every split and no source-image reuse within or across targets and replicas. Training, validation, and test splits within a replica target the same class-conditional estimated aleatoric uncertainty distribution. The dataset-specific subsections describe the substitutions used when a requested bin has insufficient eligible images. For any selected set, replacing each image's estimated aleatoric uncertainty by its bin center changes the mean by at most 0.0125 nats; departures of the selected bin counts from the ideal tilted probabilities introduce an additional discrepancy from μ.

Pool sizes and feasibility. We generated approximately class-balanced candidate pools containing 12.76 million MNIST images and 32.19 million CIFAR-10 images. We then audited the eligible supply in each class–entropy bin to determine which target distributions could support mutually disjoint, class-balanced dataset replicas.

E.2 Generated MNIST Uncertainty Family

Generator and scorer. Because the EDM release does not provide a class-conditional MNIST checkpoint, we train a generator on the 60,000-image MNIST training split following the official EDM recipe (Karras et al., 2022). Images are zero-padded from $28 \times 28 \times 1$ to $32 \times 32 \times 1$, true digit labels supply the conditioning class, and augmentation is disabled. The generator uses the DDPM++/SongUNet backbone with EDM preconditioning and EDM loss and trains for 50 Mimg. We evaluate 21 snapshots with FID-50k and select the $\mathrm{kimg} = 50000$ snapshot, which has the minimum measured FID, 0.721163. Pool sampling uses the deterministic 18-step second-order Heun sampler; generated 32×32 images are center-cropped to 28×28 before storage and scoring.

The canonical generated-image scorer contains 25 MLP-4096 teachers, five outer folds crossed with five model seeds, and uses fitted temperature $T = 2.444379$. Real MNIST reference examples use their five held-out-fold members, whereas generated images are scored by all 25 members. For each generated image, the predicted class is the argmax of the mean calibrated member probabilities and expected entropy is the mean member entropy.

Generated-image pool and targets. We generate and score approximately 12.76 million images in a class-balanced bulk phase, organizing eligible images into the shared 0.025-nat expected-entropy bins and supplementing sparse bins with targeted generation. The reference $r_{c,b}$ in Equation 8 is estimated from the cross-fitted expected entropies of the 60,000 real MNIST training images. Its class-balanced aggregate bin-center mean is $\mu_{\max} = 0.056376$ nats. The evaluated target grid is reported in Appendix F.1. The five lower targets use negative shared tilts, while the upper endpoint is the untilted real-MNIST reference. The realized pooled-supply audit filled every requested class–bin count. Of the $1{,}080{,}000$ family selections, two extreme-tail examples at $\mu_{\max}$ required borrowing from a bin three positions away because the requested bin and both immediate neighbors were empty; all remaining selections were filled at a smaller distance.

Architecture Grid, Training, and Pruning. We cross the 18 generated-family datasets and model seeds $0, 1, 2$ with two-hidden-layer ReLU MLP widths $\{32, 64, 128, 256, 512, 1024, 2048, 4096\}$. The respective learning rates are $\{0.003, 0.003, 0.003, 0.003, 0.001, 0.001, 0.001, 0.0003\}$. All conditions use AdamW with weight decay 0.1, batch size 256, cosine decay, and 200 zero-sparsity

training epochs. We apply 32 rounds of iterative global magnitude pruning, removing 20% of the remaining prunable parameters per round and finetuning for 100 epochs after every round. For the endpoint comparison in Section 8, trajectories at $\mu = 0.020000$ and $\mu = 0.056376$ whose operation-32 checkpoints still pass the fixed 2.00-percentage-point threshold use an adaptive horizon. We stop these trajectories after three consecutive checkpoints at or below the frozen threshold, subject to a maximum horizon of 60 pruning operations. This rule applies to 44 trajectories, and every selected trajectory satisfies the stopping rule before the maximum horizon.

E.3 Generated CIFAR-10 Uncertainty Family

Generated-image pool. We score a pool of $32{,}193{,}862$ CIFAR-10 images generated by the class-conditional EDM diffusion model. The calibrated canonical teacher ensemble assigns an expected-entropy score to every image. We retain an image for dataset construction only when the ensemble's predicted class agrees with the diffusion conditioning label; $30{,}968{,}601$ images satisfy this requirement.

Reference and targets. The evaluated target grid is reported in Appendix F.1. A full-pool supply audit found that three mutually disjoint replicas are feasible over $0.025293 \leq \mu \leq 0.117897$ nats when substitutions are limited to immediately adjacent entropy bins. We choose 0.030000 as a rounded low-uncertainty endpoint slightly above this feasible minimum and retain the reference bin-center mean, 0.106645 nats, as the reference endpoint rather than extending to the upper feasibility limit. This grid therefore traces progressively higher expected-entropy coordinates toward the histogram estimated on real CIFAR-10. Before materialization, we jointly audit the class–bin demand of all six targets and all three replicas against the agreeing full-pool supply; after the adjacent-bin substitutions described below, the grid has no unfilled demand. Integer class–bin counts use largest-remainder apportionment. The selected images match the requested bins exactly except for five class-9 examples at the CIFAR-reference endpoint, where the required source bin has insufficient eligible supply; these examples are drawn from an immediately adjacent 0.025-nat bin.

Architecture Grid, Training, and Pruning. We cross the 18 generated-family datasets and three model seeds with WRN-28 widen factors $k \in \{0.5, 1, 2, 4, 8\}$. Every architecture and target uses AdamW with learning rate 3×10^{-3}, weight decay 0.1, batch size 128, five warmup epochs, cosine decay, and 1,000 zero-sparsity training epochs. We apply 32 rounds of iterative global magnitude pruning, removing 20% of the remaining prunable parameters per round and finetuning for 60 epochs after every round.

E.4 Generated Image Examples at the Endpoint Targets

Figures 17 and 18 compare the lowest and original-data target distributions. All examples come from the training partition of dataset replica 0. Columns correspond to conditioning classes, and rows contain the examples nearest the 10th, 30th, 50th, 70th, 90th, and 99th percentiles of calibrated ensemble expected entropy within each class. The same selection rule is applied independently at both endpoints.



Figure 17: Diffusion-generated MNIST examples at the lowest target aleatoric uncertainty, $\mu = 0.020000$ nats (top), and the original-data target, $\mu = 0.056376$ nats (bottom). Columns show classes and rows show within-class entropy quantiles as specified above.



Figure 18: Diffusion-generated CIFAR-10 examples at the lowest target aleatoric uncertainty, $\mu = 0.030000$ nats (top), and the original-data target, $\mu = 0.106645$ nats (bottom). Columns show classes and rows show within-class entropy quantiles as specified above.

E.5 Aleatoric-Matched Evaluation

For both MNIST and CIFAR-10, every family model is evaluated on the generated test set with the same target aleatoric uncertainty and dataset seed as its training dataset, rather than on the official test set. All dense references, matching thresholds, uncertainty summaries, and absolute matching-count summaries are computed separately within each architecture size. For target μ, let $\bar{a}_{\mathrm{dense}}(\mu)$ be the mean dense matched-test accuracy across the three dataset replicas crossed with the three model seeds. The standard accuracy-matching threshold is

$$T_{\mathrm{SD}}(\mu) = \bar{a}_{\mathrm{dense}}(\mu) - \sigma_{\mathrm{dense}}(\mu), \tag{9}$$

where $\sigma_{\mathrm{dense}}(\mu)$ is the population standard deviation across those nine runs. For the main architecture-size analysis, let μ_{orig} denote the original-data target for the dataset and define the fixed-original-dataset-SD threshold

$$T_{\mathrm{fixed}}(\mu) = \bar{a}_{\mathrm{dense}}(\mu) - \sigma_{\mathrm{dense}}(\mu_{\mathrm{orig}}). \tag{10}$$

This rule computes one tolerance for each architecture size from its nine original-target dense runs and reuses that tolerance across all six generated-family targets while retaining each target's own dense mean. The fixed-drop threshold is

$$T_{\delta}(\mu) = \bar{a}_{\mathrm{dense}}(\mu) - \delta, \tag{11}$$

where $\delta \in \{0.10, 0.25, 0.50, 1.00, 1.50, 2.00\}$ is measured in absolute percentage points. For all three threshold rules, we linearly interpolate matched-test accuracy and retained parameter count between adjacent checkpoints within each run, including the zero-sparsity-to-first-pruned segment. The pruned network size is the smallest count among measured positive-sparsity checkpoints whose accuracy is strictly greater than the threshold and interpolated threshold crossings; if neither exists, we retain the zero-sparsity count. Crossings include equality at segment endpoints, following the interpolation rule used for the natural-data label-noise figures. Interpolation is linear in retained parameter count, equivalently in sparsity within a fixed architecture; no checkpoint weights are interpolated or evaluated, and there is no extrapolation. Because architecture size and evaluation split are fixed within each dataset, let $\widehat{M}_{\gamma}^{\mathrm{test}}(\mu)$ abbreviate the nine-run mean empirical matching count at target μ under criterion γ. The main-text endpoint table and Appendix F.1 report this absolute count under the fixed-original-dataset-SD rule; Appendix F.6 compares the condition-specific and fixed-drop criteria. The estimated original-data uncertainty is 0.056376 for MNIST and 0.106645 for CIFAR-10; dashed guides mark these coordinates without normalizing the plotted counts to them.

E.6 Endpoint Finetuning Across the Architecture Grid

We use two-hidden-layer MNIST MLPs with widths $\{32, 64, 128, 256, 512, 1024, 2048, 4096\}$ and CIFAR-10 WRN-28 models with widen factors $\{0.5, 1, 2, 4, 8\}$. The reduced- and original-aleatoric-uncertainty generated endpoints have target uncertainties of 0.020000 and 0.056376 nats for MNIST and 0.030000 and 0.106645 nats for CIFAR-10, respectively. For each architecture and source endpoint, we restore the zero-sparsity checkpoint and all 32 positive-sparsity MNIST checkpoints or 22 positive-sparsity CIFAR-10 checkpoints, crossed with three generated-dataset replicas and three model seeds. The transfer analysis therefore uses the first 22 CIFAR-10 pruning rounds rather than the full 32-round family trajectory. Each checkpoint is evaluated on its paired original-aleatoric-uncertainty generated test set, finetuned on the corresponding training set with its pruning mask fixed and no additional pruning, and evaluated again on that same test set. The MNIST finetuning phase lasts 200 epochs, with the width-specific AdamW learning rates used in the architecture study, batch size 256, and weight decay 0.1; the CIFAR-10 phase lasts $1{,}000$ epochs with AdamW, initial learning rate 3×10^{-3}, five warmup epochs, batch size 128, and weight decay 0.1. In both cases, finetuning lasts as long as the source model's zero-sparsity training phase.

We pair the two source conditions by architecture, exact pruning round, dataset replica, and model seed, yielding nine pairs at every checkpoint. The endpoint gap is original-trained minus reduced-trained accuracy, in percentage points, on the common original-aleatoric-uncertainty generated test set; a smaller gap favors the reduced-aleatoric-uncertainty initialization. We report the gap at zero sparsity, the mean of checkpoint-mean gaps across each trajectory, the number of checkpoints whose gap decreases after finetuning, and the number at which the finetuned reduced-aleatoric-uncertainty

mean equals or exceeds the finetuned original-aleatoric-uncertainty mean. To measure curve separation without cancellation between positive and negative gaps, we also average the absolute difference between the two nine-run condition means at each checkpoint, first across checkpoints and then equally across architectures within each dataset. These gap statistics pair exact rounds without interpolation and do not assume identical retained parameter counts across source conditions. Figure lines average the nine completed seed cells, while bands show their observed minima and maxima rather than confidence intervals. The pre-finetuning dashed guides mark separate source-distribution matching counts under the condition-specific dense matched-test mean-minus-population-standard-deviation rule; those guides do not select the checkpoints included in this experiment.

E.7 Exponential Tilting as the Minimum-KL Uncertainty Shift

We first justify the ideal input-distribution tilt in Section 8, then its class-balanced histogram counterpart used for both MNIST and CIFAR-10. Both follow from the KL decomposition underlying information projections (Csiszár, 1975; Csiszár & Shields, 2004).

E.7.1 Ideal Input-Distribution Tilt

Let $h(x) = H(p(\cdot \mid x))$ be bounded and measurable, and retain the conditional label distribution $p(\cdot \mid x)$ while varying the input marginal. For finite $\lambda \in \mathbb{R}$, define

$$Z(\lambda) = \mathbb{E}_{X \sim P_X}[e^{\lambda h(X)}], \qquad \frac{dP_X^\lambda}{dP_X}(x) = \frac{e^{\lambda h(x)}}{Z(\lambda)}. \tag{12}$$

Boundedness gives $0 < Z(\lambda) < \infty$, and the positive density makes P_X^λ and P_X mutually absolutely continuous. Thus the tilt preserves input null sets and the set of Bayes decision rules. Suppose λ achieves $\mathbb{E}_{P_X^\lambda}[h] = \alpha$. For every alternative input marginal $Q_X \ll P_X$ with $\mathbb{E}_{Q_X}[h] = \alpha$ and finite forward KL divergence, expanding the log density gives

$$D_{\mathrm{KL}}(Q_X \| P_X) = D_{\mathrm{KL}}(Q_X \| P_X^\lambda) + \lambda\alpha - \log Z(\lambda) \tag{13}$$

$$= D_{\mathrm{KL}}(Q_X \| P_X^\lambda) + D_{\mathrm{KL}}(P_X^\lambda \| P_X). \tag{14}$$

Nonnegativity of KL divergence proves that P_X^λ uniquely minimizes $D_{\mathrm{KL}}(Q_X \| P_X)$ over input marginals with this mean; candidates with infinite divergence cannot improve it. For joint distributions $Q_{XY} = Q_X p(\cdot \mid x)$ sharing the fixed conditional labels, $D_{\mathrm{KL}}(Q_{XY} \| P_{XY}) = D_{\mathrm{KL}}(Q_X \| P_X)$, so the same minimum holds for the uncertainty modifiers considered here. If h is not constant almost surely, the tilted mean has derivative $\mathrm{Var}_{P_X^\lambda}(h) > 0$ and tends to the essential lower and upper bounds of h as $\lambda \to -\infty$ and $\lambda \to +\infty$. Every target strictly between those bounds therefore has a unique finite tilt. If h is constant almost surely, only that mean is attainable and every tilt equals P_X.

E.7.2 Class-Balanced Histogram Tilt

For the implemented construction, the corresponding guarantee concerns the ideal class–entropy-bin probabilities with class proportions fixed. It does not assert minimum KL divergence between generated image distributions or between finite datasets after integer allocation and bin substitutions.

Setup and support. Let $c \in \{1, \ldots, C\}$ index classes, let $b \in \{1, \ldots, B\}$ index bins with centers $h_b \in \mathbb{R}$, and fix class probabilities $\pi_c > 0$ with $\sum_c \pi_c = 1$. Our construction uses $\pi_c = 1/C$. Let $r_{c,b} \geq 0$, with $\sum_b r_{c,b} = 1$, be the reference class-conditional bin probabilities and define $r(c,b) = \pi_c r_{c,b}$ and $S_r = \{(c,b) : r(c,b) > 0\}$. For an alternative distribution $p(c,b) = \pi_c p_{c,b}$, the feasible family for target mean μ is

$$\mathcal{P}_\mu = \left\{ p : p_{c,b} \geq 0, \quad \sum_b p_{c,b} = 1 \text{ for every } c, \quad \sum_c \pi_c \sum_b p_{c,b} h_b = \mu \right\}. \tag{15}$$

Define

$$D_{\mathrm{KL}}(p \| r) = \sum_{c,b} p(c,b) \log \frac{p(c,b)}{r(c,b)}, \tag{16}$$

using natural logarithms, $0\log(0/v) = 0$, and $u\log(u/0) = +\infty$ for $u > 0$. Any candidate assigning positive mass outside S_r has infinite forward KL divergence, so a finite optimum must lie in

$$\mathcal{P}_\mu^{(r)} = \{p \in \mathcal{P}_\mu : \mathrm{supp}(p) \subseteq S_r\}. \tag{17}$$

Tilt and existence. For finite $\lambda \in \mathbb{R}$, define

$$Z_c(\lambda) = \sum_b r_{c,b} e^{\lambda h_b}, \qquad q_{\lambda,c,b} = \frac{r_{c,b} e^{\lambda h_b}}{Z_c(\lambda)}, \qquad q_\lambda(c,b) = \pi_c q_{\lambda,c,b}. \tag{18}$$

The tilted distribution has exactly the reference support S_r and aggregate mean $\bar{h}(\lambda) = \sum_c \pi_c \sum_b q_{\lambda,c,b} h_b$. The attainable means with finite forward KL divergence form the interval $[\mu_-, \mu_+]$, where

$$\mu_- = \sum_c \pi_c \min_{b: r_{c,b} > 0} h_b, \qquad \mu_+ = \sum_c \pi_c \max_{b: r_{c,b} > 0} h_b. \tag{19}$$

Differentiation gives

$$\bar{h}'(\lambda) = \sum_c \pi_c \,\mathrm{Var}_{B \sim q_{\lambda,c,\cdot}}(h_B) \geq 0. \tag{20}$$

If at least one class has positive reference mass at two distinct bin centers, this derivative is strictly positive for every finite λ. Moreover, $\bar{h}(\lambda)$ tends to μ_- and μ_+ as λ tends to $-\infty$ and $+\infty$, respectively. Consequently, every interior target has a unique finite λ satisfying $\bar{h}(\lambda) = \mu$. If $\mu_- = \mu_+$, the only attainable target has the unique KL minimizer r, and every λ produces this same distribution.

Minimum-KL property. Fix a finite λ with $\bar{h}(\lambda) = \mu$. For every $p \in \mathcal{P}_\mu^{(r)}$, the fixed class probabilities and mean imply

$$D_{\mathrm{KL}}(p\|r) - D_{\mathrm{KL}}(p\|q_\lambda) = \sum_{(c,b) \in S_r} p(c,b) \log \frac{q_\lambda(c,b)}{r(c,b)} \tag{21}$$

$$= \lambda\mu - \sum_c \pi_c \log Z_c(\lambda) \tag{22}$$

$$= D_{\mathrm{KL}}(q_\lambda\|r). \tag{23}$$

Thus $D_{\mathrm{KL}}(p\|r) = D_{\mathrm{KL}}(p\|q_\lambda) + D_{\mathrm{KL}}(q_\lambda\|r)$. Nonnegativity of KL divergence, with equality only when $p = q_\lambda$, gives

$$q_\lambda = \arg\min_{p \in \mathcal{P}_\mu} D_{\mathrm{KL}}(p\|r), \tag{24}$$

where candidates outside S_r cannot improve the finite objective.

At an endpoint μ_- or μ_+, every finite-KL feasible distribution must restrict each class to its minimum- or maximum-center reference bins, respectively. Let $E_c^\pm$ denote these extremal bins and $a_c^\pm = \sum_{b \in E_c^\pm} r_{c,b}$. The limiting tilt is $q_{\pm,c,b} = r_{c,b}\mathbf{1}\{b \in E_c^\pm\}/a_c^\pm$, and every endpoint-feasible p satisfies

$$D_{\mathrm{KL}}(p\|r) = D_{\mathrm{KL}}(p\|q_\pm) - \sum_c \pi_c \log a_c^\pm. \tag{25}$$

This proves uniqueness of the endpoint minimizer as well, although an endpoint tilt can reduce the reference support. Outside $[\mu_-, \mu_+]$, no finite-KL feasible distribution exists.

A shared λ follows from imposing one aggregate mean constraint; separate classwise means would generally require separate λ_c. The result concerns forward divergence $D_{\mathrm{KL}}(p\|r)$, not reverse divergence $D_{\mathrm{KL}}(r\|p)$.

F Natural-Data Uncertainty-Reduction Results (Extended)

This appendix reports the full uncertainty-target sweep, endpoint finetuning across the MNIST and CIFAR-10 architecture grids, additional diagnostics for the generated uncertainty families, and possible sources of nonmonotone trends. The endpoint-finetuning experiment evaluates both source conditions on the same paired original-aleatoric-uncertainty generated test set, before and after equal

finetuning on the corresponding training set. By contrast, the uncertainty-family diagnostics in Appendix F.6 use aleatoric-matched generated test sets: a model trained at target μ and dataset replica s is evaluated at that same μ and s.

F.1 Results Across the Full Uncertainty-Target Family

For each dataset, we also evaluated four intermediate targets between the two endpoints compared in Section 8. The selected target bin-center means are

$$\text{MNIST:} \quad \mu \in \{0.020, 0.026, 0.032, 0.038, 0.044, 0.056376\},$$
$$\text{CIFAR-10:} \quad \mu \in \{0.030, 0.045, 0.060, 0.075, 0.090, 0.106645\}.$$

The lowest CIFAR-10 target is a rounded value slightly above the minimum supported by the audited image pool; the upper endpoints correspond to the original datasets' reference histograms. These coordinates use bin-center estimates; Appendix F.3 compares them with the realized mean estimated aleatoric uncertainty of the selected images. Figure 19 shows the pruned network sizes at all six target levels. Every displayed architecture-size curve has at least one reversal between adjacent targets, although the endpoint comparison generally shows smaller pruned network sizes at lower aleatoric uncertainty. Appendix F.2 discusses the reversals and the variation across dataset replicas.

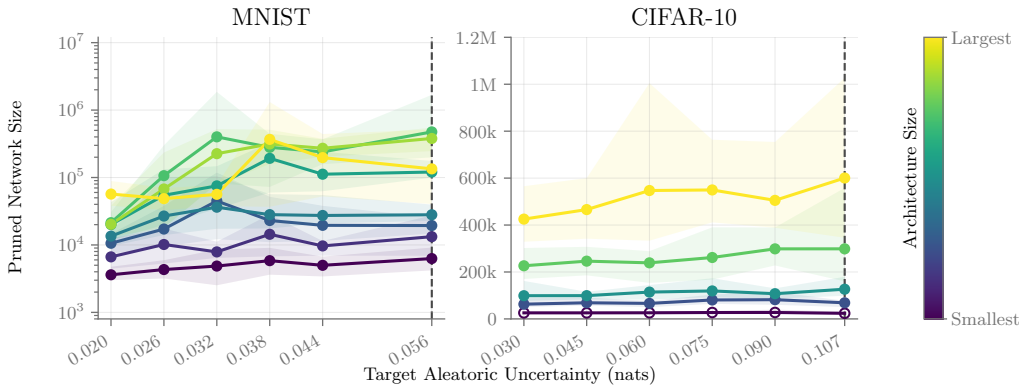

Figure 19: Target aleatoric uncertainty versus pruned network size for MNIST (left) and CIFAR-10 (right), using distribution shaping. The horizontal axes are linear; the MNIST vertical axis is logarithmic and the CIFAR-10 vertical axis is linear. Each curve corresponds to an architecture size, and each point averages three dataset replicas crossed with three model seeds. Dashed vertical black lines mark the estimated aleatoric uncertainty of the original datasets. Shaded bands span the minimum and maximum pruned network sizes across the nine runs at each target. Filled markers indicate that every zero-sparsity replicate memorizes; open markers indicate that at least one does not.

F.2 Possible Sources of Nonmonotonicity

Variation across the three generated dataset replicas at each target could contribute to the reversals. Across the displayed architecture–target cells, the median coefficient of variation of the three dataset-replica mean pruned network sizes is 34.9% for MNIST and 11.4% for CIFAR-10, with each replica mean averaging three model seeds. For 17 of the 22 adjacent-target reversals, removing one dataset replica from each target changes the direction of the mean difference in at least one of the nine possible removal combinations; five reversals retain their direction in all nine combinations. More independently generated dataset replicas would be needed to assess which reversals persist beyond these samples.

Error in the estimated aleatoric uncertainty is another possible source, particularly when zero-sparsity test accuracies are near saturation. For generated MNIST, an ensemble trained on diffusion-generated data gives uncertainty estimates that differ in magnitude from those of the original-data

Table 7: Target bin-center coordinates and realized mean estimated aleatoric uncertainty for MNIST, in nats. Generated entries give the mean across three dataset replicas, with the observed replica minimum and maximum in brackets; each replica contains $45{,}000$ training, $5{,}000$ validation, and $10{,}000$ test examples. The original row reports the bin-center and unbinned means of the full $60{,}000$-image original training reference. All values are rounded to six decimal places.

Source	Bin-Center μ	Training	Validation	Test
Original	0.056376	0.049595	–	–
Generated	0.020000	0.012619 $[0.012610, 0.012630]$	0.012874 $[0.012819, 0.012908]$	0.012302 $[0.012276, 0.012342]$
Generated	0.026000	0.018789 $[0.018697, 0.018888]$	0.018567 $[0.017814, 0.019490]$	0.018071 $[0.018043, 0.018105]$
Generated	0.032000	0.024856 $[0.024659, 0.025035]$	0.024406 $[0.022533, 0.026312]$	0.023994 $[0.023948, 0.024069]$
Generated	0.038000	0.030843 $[0.030600, 0.031016]$	0.031709 $[0.030406, 0.033794]$	0.029708 $[0.029690, 0.029734]$
Generated	0.044000	0.036985 $[0.036920, 0.037046]$	0.036951 $[0.036641, 0.037342]$	0.036227 $[0.036181, 0.036287]$
Generated	0.056376	0.050150 $[0.050035, 0.050372]$	0.049914 $[0.048023, 0.050890]$	0.048375 $[0.048356, 0.048403]$

ensemble but preserve the ordering of target-level means (Appendix F.4). This comparison does not establish the true aleatoric uncertainty of the generated datasets. Our exponential-tilting procedure also matches target statistics of discretized aleatoric uncertainty bins rather than unbinned scores; the realized unbinned means differ from the nominal targets (Appendix F.3).

F.3 Bin-Center Approximation and Realized Aleatoric Uncertainty

Distribution shaping sets each target μ from the expected bin-center mean of a tilted histogram rather than from the individual images' estimated aleatoric uncertainty scores. Integer apportionment, substitutions for bins with insufficient supply, and variation of image scores within bins can each shift the realized mean. Consequently, the target coordinate μ can differ from the realized mean estimated aleatoric uncertainty of the images selected into those bins. The horizontal coordinates of Figure 19, including its original-dataset guides, use the bin-center values; its shaded bands describe variation in pruned network size and do not represent this horizontal-coordinate approximation. Tables 7 and 8 show the target bin-center value alongside the realized split means, averaged over the three dataset replicas, with their observed minimum–maximum ranges. The discrepancy between bin-center and unbinned means is a methodological limitation of the displayed uncertainty coordinate; neither value resolves error in the ensemble's estimate of the true aleatoric uncertainty.

Table 8: Target bin-center coordinates and realized mean estimated aleatoric uncertainty for CIFAR-10, in nats. Generated entries give the mean across three dataset replicas, with the observed replica minimum and maximum in brackets; each replica contains $45{,}000$ training, $5{,}000$ validation, and $10{,}000$ test examples. The original row reports the bin-center and unbinned means of the full $50{,}000$-image original training reference. All values are rounded to six decimal places.

Source	Bin-Center μ	Training	Validation	Test
Original	0.106645	0.105902	–	–
Generated	0.030000	0.029332 $[0.029314, 0.029363]$	0.029520 $[0.029341, 0.029680]$	0.028921 $[0.028866, 0.029017]$
Generated	0.045000	0.044238 $[0.043995, 0.044404]$	0.043724 $[0.042190, 0.046032]$	0.043544 $[0.043504, 0.043613]$
Generated	0.060000	0.059106 $[0.058890, 0.059339]$	0.059927 $[0.057530, 0.061762]$	0.058134 $[0.058105, 0.058153]$
Generated	0.075000	0.074487 $[0.074265, 0.074721]$	0.072761 $[0.070914, 0.074701]$	0.073179 $[0.073154, 0.073194]$
Generated	0.090000	0.089353 $[0.088950, 0.089767]$	0.092221 $[0.088289, 0.095948]$	0.087892 $[0.087853, 0.087963]$
Generated	0.106645	0.106283 $[0.106075, 0.106443]$	0.104307 $[0.102748, 0.106146]$	0.104037 $[0.103996, 0.104101]$

All unbinned means are equal-class averages of H_{exp}; for the exactly class-balanced generated splits, these equal the ordinary sample means. The original-reference rows summarize the full original training populations used to construct the reference histograms, comprising $60{,}000$ MNIST and $50{,}000$ CIFAR-10 images. For MNIST, equal class weighting gives an unbinned reference mean of 0.049595 nats; weighting by the original class frequencies instead gives 0.049262 nats. We compute the generated MNIST split means directly from the published dataset arrays and recover the generated CIFAR-10 split means from the original training runs, verifying exact agreement across the three model seeds for each dataset replica. These are dataset statistics, so no averaging across model seeds is performed. The CIFAR-10 reference uses the canonical GPU-scored coordinate from the original family-construction audit. The generated endpoint with the reference bin-center coordinate need not reproduce the reference's unbinned mean.

F.4 MNIST Ensemble-Dependence Diagnostic

The generated MNIST family was constructed using estimated aleatoric uncertainty from the calibrated ensemble trained on real MNIST, so its numerical uncertainty coordinates may depend on that scorer's training distribution. To assess this dependence, we trained another five-fold, five-seed MLP ensemble on the $50{,}000$ development images of the generated MNIST dataset at the original-data target $\mu = 0.056376$ nats, using the same architecture and training protocol as the real-MNIST ensemble and fitting a separate temperature from out-of-fold predictions. The real-MNIST and generated-trained ensemble temperatures are 2.444379 and 1.905134, respectively. We applied all 25 generated-trained members to the $10{,}000$ matched-test images in each of the 18 selected MNIST family datasets and compared their mean member entropy with the original scorer values retained in the dataset files. These $180{,}000$ test images were held out from the generated-trained ensemble.

Figure 20 shows the mean estimated aleatoric uncertainty under both ensembles at each selection target. The six target-level means have the same ordering under both scorers (Spearman $\rho = 1.000$; Pearson $r = 0.999$), although the generated-trained ensemble's means are 0.708–0.800 times the real-MNIST ensemble's means. At the lowest and original-data selection targets, the original scorer's realized matched-test means are 0.012302 and 0.048375 nats, while the generated-trained scorer's means are 0.009838 and 0.034229 nats. Across paired test images, the two estimated aleatoric uncertainties have Spearman correlation $\rho = 0.876$. For the generated-trained ensemble's own $50{,}000$ development images, a separate five-held-out-member-per-image evaluation gives mean expected entropy 0.036275 nats, compared with 0.050121 nats from the real-MNIST scorer.

The lower values from the generated-trained scorer are consistent with the real-MNIST scorer responding to distribution shift on diffusion-generated images. Because neither ensemble supplies ground-truth conditional entropy, this comparison does not establish which absolute estimate is closer to the data's aleatoric uncertainty. The aggregate ordering of the six generated MNIST uncertainty levels is preserved under this change of scorer, while their numerical spacing depends on the scorer.

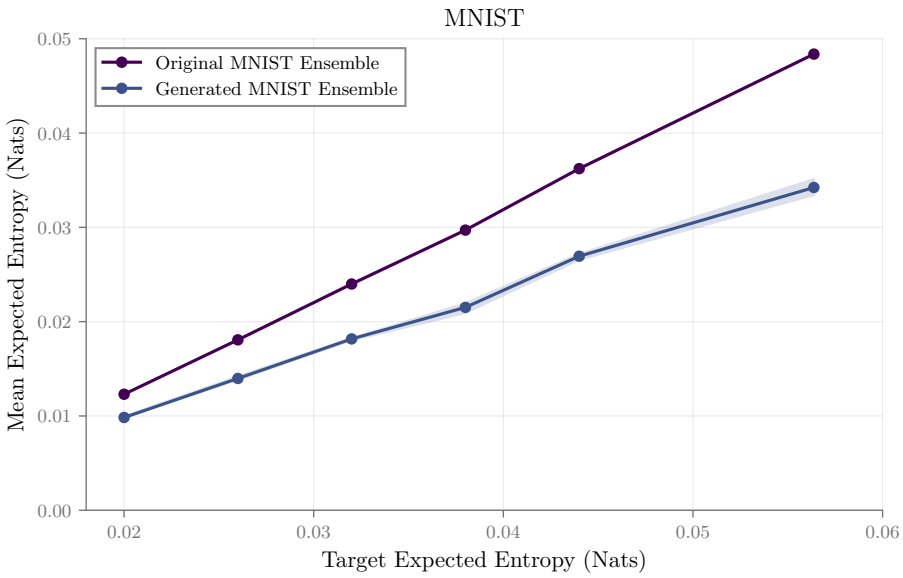

Figure 20: Mean estimated aleatoric uncertainty on the matched-test partitions of the generated MNIST family under the real-MNIST and generated-trained ensembles. Each point averages three mutually disjoint dataset replicas of 10,000 test images; shaded bands span the minimum and maximum replica means. The horizontal coordinate is the selection target defined using the original ensemble's class-conditional histogram bin centers, while plotted vertical values are means of unbinned per-image expected entropy.

F.5 Endpoint Finetuning Across the Architecture Grid

Smaller pruned network sizes on reduced-aleatoric-uncertainty generated data do not establish whether those models retain accuracy on an original-aleatoric-uncertainty distribution. We therefore ask whether finetuning narrows the original-aleatoric-uncertainty test-accuracy gap between models trained at the reduced- and original-aleatoric-uncertainty generated endpoints throughout the pruning trajectory and across initial architecture sizes. Both source conditions receive the same original-aleatoric-uncertainty finetuning phase, making this an endpoint-initialization comparison rather than a comparison between finetuned and unfinetuned models.

The architecture grids, fixed-mask finetuning protocol, exact-round pairing, and gap statistics are specified in Appendix E.6.

Tables 9 and 10 summarize the complete paired gaps, Figure 21 compares trajectory-mean gaps across architectures, and Figures 22–25 show all trajectories. At zero sparsity, equal finetuning decreases the original-trained-minus-reduced-trained gap for all 13 architectures by 0.384–0.868 percentage points. The post-finetuning zero-sparsity gap is nonpositive for MNIST widths 512, 1024, and 4096 and CIFAR-10 widen factor 2. The signed trajectory-mean gap decreases for every architecture, and its equal-architecture average changes from 0.573 to 0.132 percentage points across the eight MNIST widths and from 0.780 to 0.347 percentage points across the five CIFAR-10 widen factors. For larger architectures, the same average changes from 0.518 to -0.090 percentage points across MNIST widths 512–4096 and from 0.879 to 0.265 percentage points across CIFAR-10 widen factors 4 and 8. Negative gaps indicate higher original-aleatoric-uncertainty test accuracy for models pruned on reduced-aleatoric-uncertainty data. After finetuning, this signed gap is negative for five

MNIST widths and positive for all five CIFAR-10 widen factors. The average absolute checkpoint-mean gap decreases from 0.761 to 0.457 percentage points on MNIST and from 0.821 to 0.416 percentage points on CIFAR-10.

The gap decreases at every checkpoint for MNIST widths 512, 1024, 2048, and 4096 and CIFAR-10 widen factors 2, 4, and 8. At smaller architectures, the improvement is less uniform: the maximum absolute checkpoint-mean gap increases after finetuning for MNIST widths 32, 64, and 128 and CIFAR-10 widen factors 0.5 and 1. Thus, equal finetuning consistently narrows the zero-sparsity and trajectory-average source-condition differences, but does not guarantee gap closure at every sparsity or dominance of the reduced-aleatoric-uncertainty initialization. Because this experiment compares fixed pruning rounds rather than newly selected checkpoints under the matching criterion, these results alone do not establish a parameter-count reduction relative to an original-aleatoric-uncertainty-trained model selected under that criterion.

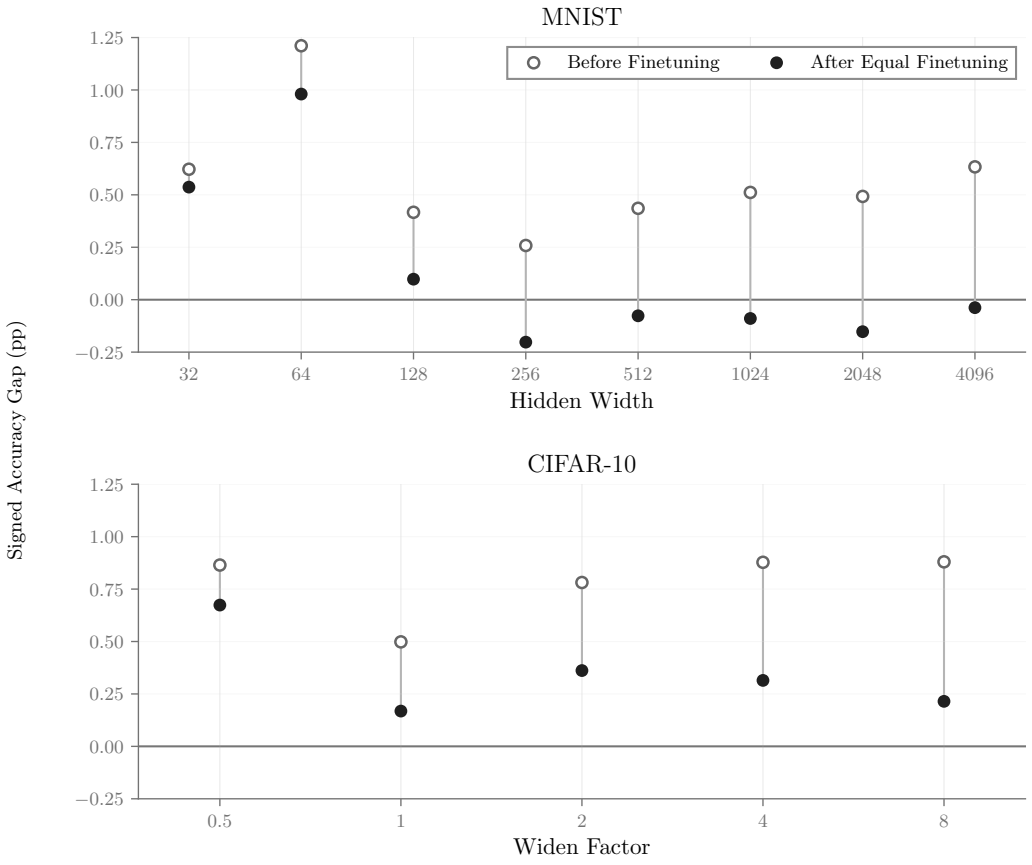

Figure 21: Signed original-trained-minus-reduced-trained original-aleatoric-uncertainty test-accuracy gap, averaged over exact pruning-round checkpoints for each architecture. Hollow points show values before finetuning, filled points show values after equal finetuning, and gray segments connect values for the same architecture. Each checkpoint mean uses nine paired dataset-replica/model-seed cells; the MNIST and CIFAR-10 trajectory means average 33 and 23 checkpoints, respectively. Values below zero indicate higher average accuracy for reduced-aleatoric-uncertainty-trained models.

Table 9: MNIST original-trained-minus-reduced-trained test-accuracy gaps in percentage points, with nine exact seed pairs per checkpoint. The trajectory-mean gap averages the 33 checkpoint means; "Gap reduced" counts checkpoints with a smaller gap after finetuning; and "Reduced after $\geq$ original" counts nonpositive post-finetuning gaps. The maximum absolute gap is selected separately before and after finetuning and may occur at different checkpoints.

MNIST width	Zero-sparsity gap before $\rightarrow$ after	Trajectory-mean gap before $\rightarrow$ after	Gap reduced	Reduced after $\geq$ original	Maximum absolute gap before $\rightarrow$ after
32	$0.441 \rightarrow 0.052$	$0.622 \rightarrow 0.537$	23/33	17/33	$3.622 \rightarrow 3.906$
64	$0.599 \rightarrow 0.098$	$1.211 \rightarrow 0.981$	24/33	10/33	$8.570 \rightarrow 8.882$
128	$0.770 \rightarrow 0.078$	$0.417 \rightarrow 0.098$	25/33	9/33	$2.499 \rightarrow 2.984$
256	$0.844 \rightarrow 0.046$	$0.258 \rightarrow -0.202$	29/33	10/33	$3.324 \rightarrow 3.241$
512	$0.793 \rightarrow -0.016$	$0.436 \rightarrow -0.077$	33/33	24/33	$1.117 \rightarrow 0.817$
1024	$0.852 \rightarrow -0.016$	$0.511 \rightarrow -0.090$	33/33	25/33	$1.094 \rightarrow 0.773$
2048	$0.868 \rightarrow 0.062$	$0.492 \rightarrow -0.153$	33/33	20/33	$1.247 \rightarrow 0.739$
4096	$0.630 \rightarrow -0.099$	$0.633 \rightarrow -0.038$	33/33	25/33	$0.724 \rightarrow 0.162$

Table 10: CIFAR-10 original-trained-minus-reduced-trained test-accuracy gaps in percentage points, with nine exact seed pairs per checkpoint. Column definitions match Table 9, with 23 checkpoints per architecture.

CIFAR-10 widen factor	Zero-sparsity gap before $\rightarrow$ after	Trajectory-mean gap before $\rightarrow$ after	Gap reduced	Reduced after $\geq$ original	Maximum absolute gap before $\rightarrow$ after
0.5	$0.427 \rightarrow 0.042$	$0.865 \rightarrow 0.674$	18/23	3/23	$4.896 \rightarrow 5.336$
1	$0.741 \rightarrow 0.076$	$0.498 \rightarrow 0.168$	22/23	7/23	$0.970 \rightarrow 1.257$
2	$0.702 \rightarrow -0.024$	$0.781 \rightarrow 0.361$	23/23	2/23	$1.243 \rightarrow 1.084$
4	$0.751 \rightarrow 0.169$	$0.878 \rightarrow 0.315$	23/23	2/23	$1.098 \rightarrow 0.824$
8	$0.734 \rightarrow 0.024$	$0.880 \rightarrow 0.215$	23/23	1/23	$0.951 \rightarrow 0.664$

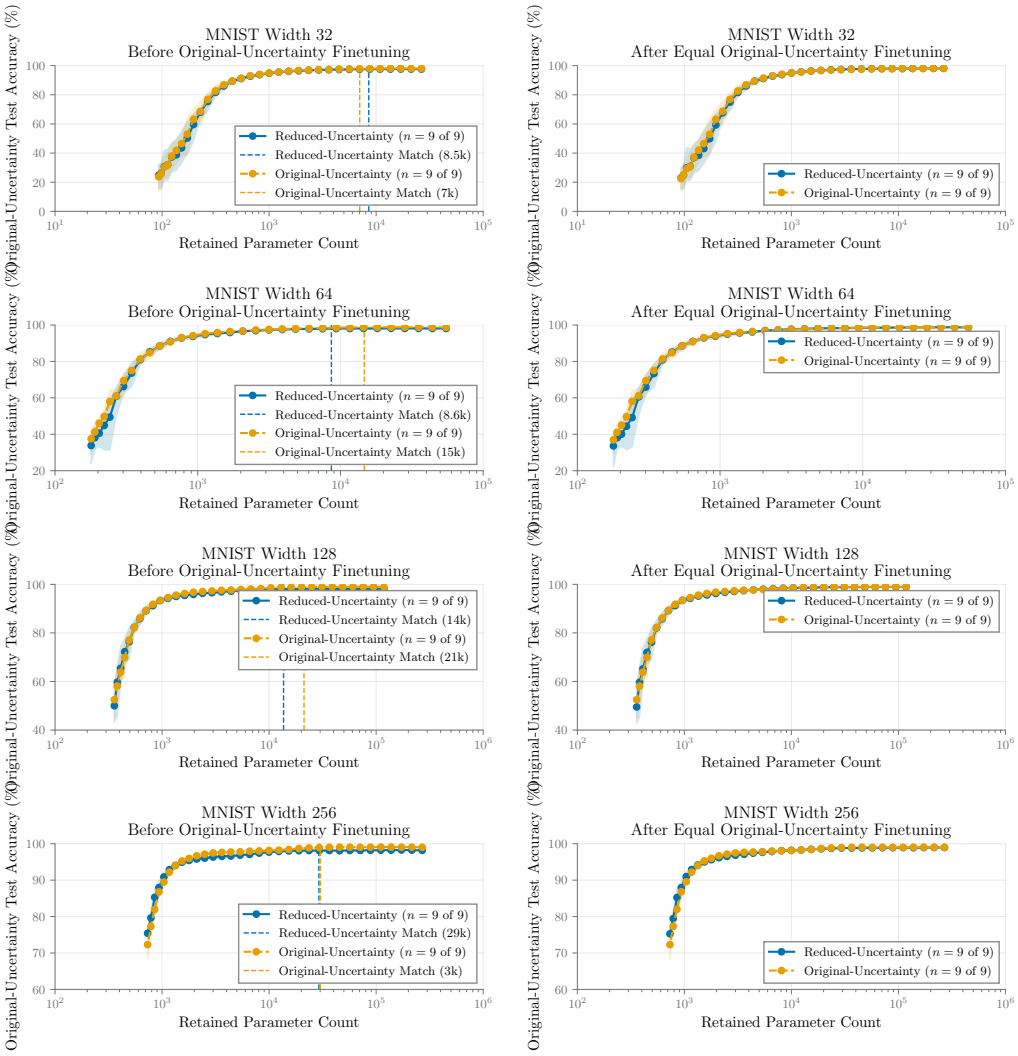

Figure 22: Original-aleatoric-uncertainty test accuracy for MNIST widths 32–256, before and after equal original-aleatoric-uncertainty finetuning. Blue shows reduced-aleatoric-uncertainty-trained checkpoints and orange shows original-aleatoric-uncertainty-trained checkpoints.

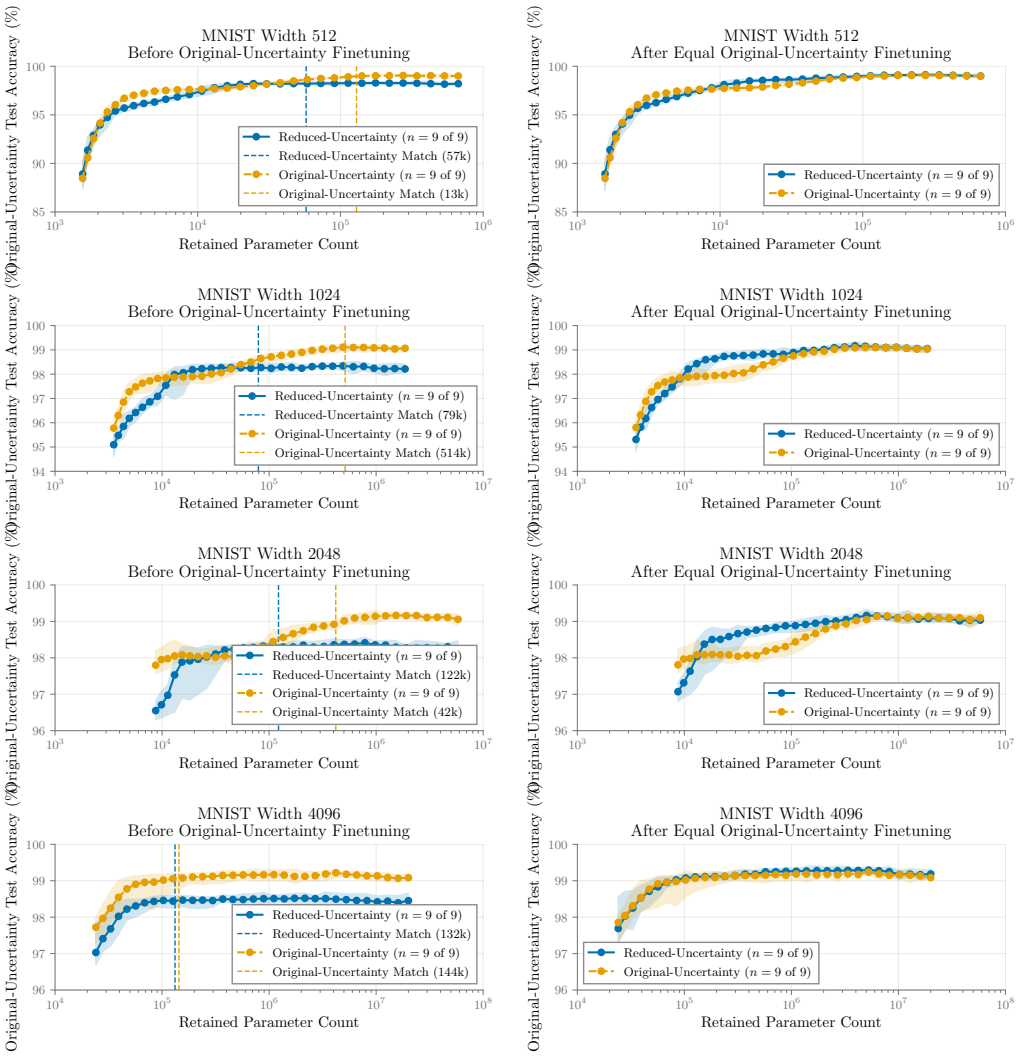

Figure 23: Original-aleatoric-uncertainty test accuracy for MNIST widths 512–4096, using the same conventions as Figure 22.

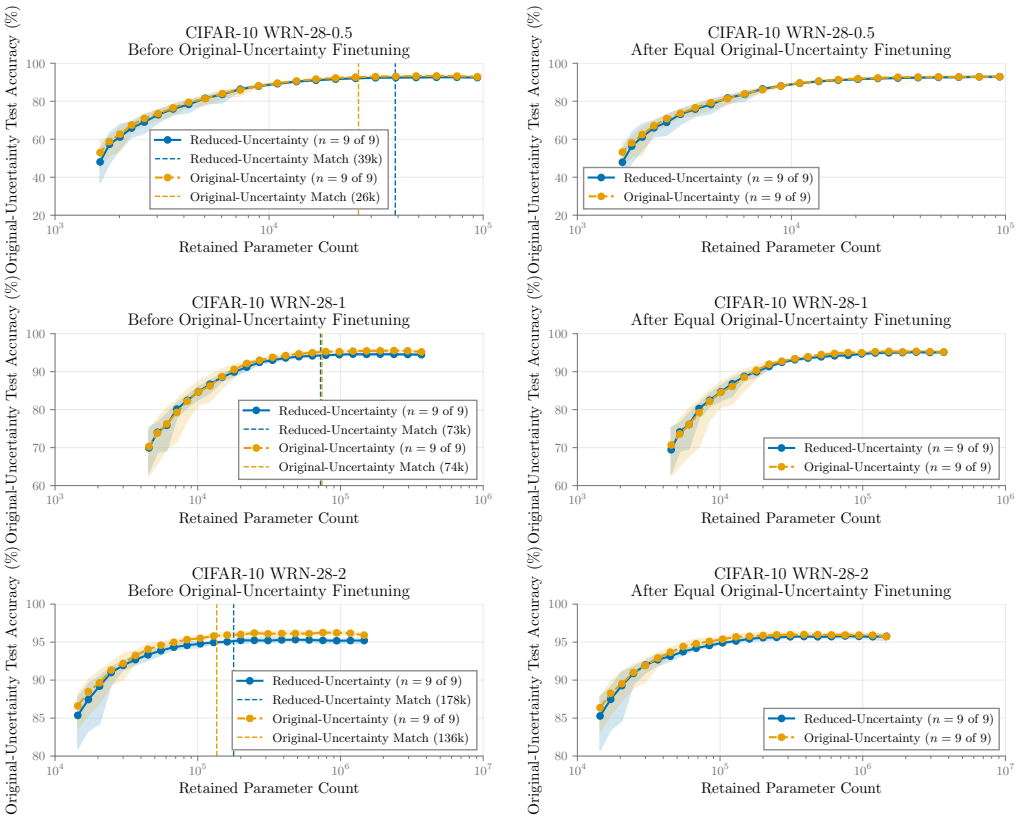

Figure 24: Original-aleatoric-uncertainty test accuracy for CIFAR-10 WRN-28 widen factors 0.5, 1, and 2, using the same conventions as Figure 22.

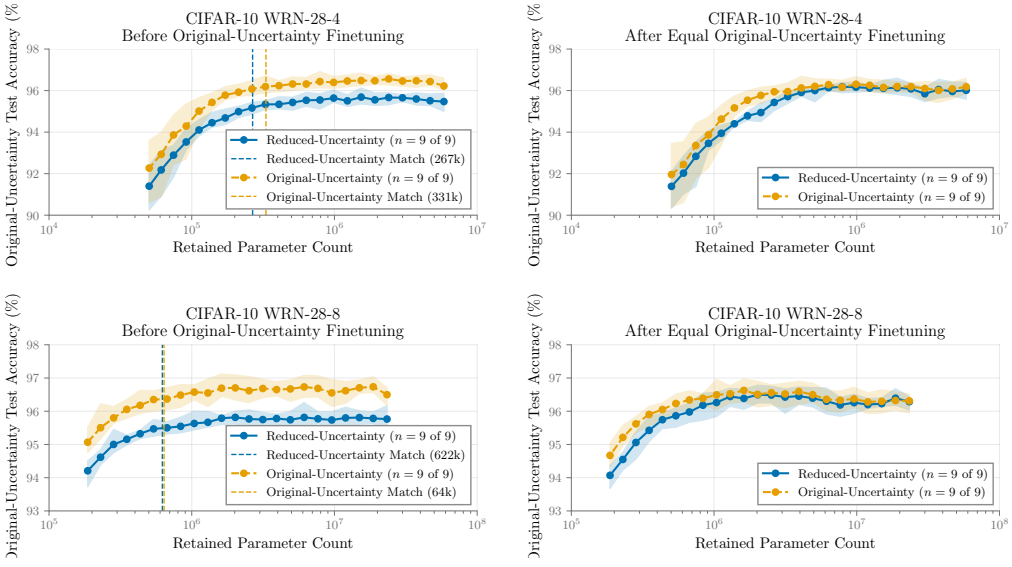

Figure 25: Original-aleatoric-uncertainty test accuracy for CIFAR-10 WRN-28 widen factors 4 and 8, using the same conventions as Figure 22.

F.6 Matching-Criterion Saturation Diagnostics

The condition-specific criterion in Equation (9) subtracts each target's own dense test-accuracy population standard deviation. As aleatoric uncertainty decreases, dense test accuracy approaches the 100% ceiling and dense test error falls (Figure 27). The dense test-accuracy population standard deviation also generally contracts at these lower-aleatoric-uncertainty targets (Figure 27). Because the standard criterion subtracts this target-specific population standard deviation, a smaller value places the threshold closer to the dense mean and permits a smaller accuracy loss. The standard criterion is therefore systematically stricter at lower aleatoric uncertainty, and its pruned network sizes mix the effect of changing aleatoric uncertainty with the effect of changing tolerance (Figure 26).

The main-text fixed-original-dataset-SD criterion removes this source of variation by computing $\sigma_{\text{dense}}(\mu_{\text{orig}}, n)$ once for each architecture size and reusing it across all six target aleatoric uncertainty levels. We also use the fixed absolute accuracy drops in Equation (11), spanning 0.10, 0.25, 0.50, 1.00, 1.50, and 2.00 percentage points. Table 11 reports the endpoint reduction under every tolerance, and Figure 28 illustrates the full target sweep for the 1.00-percentage-point criterion. Every diagnostic uses three mutually disjoint dataset replicas crossed with three model seeds per architecture-size and target cell; lines show nine-trajectory means and bands span their observed minima and maxima. These diagnostics use the same per-run interpolation rule as the main-text endpoint table and the full-family figure, as described in Appendix E.

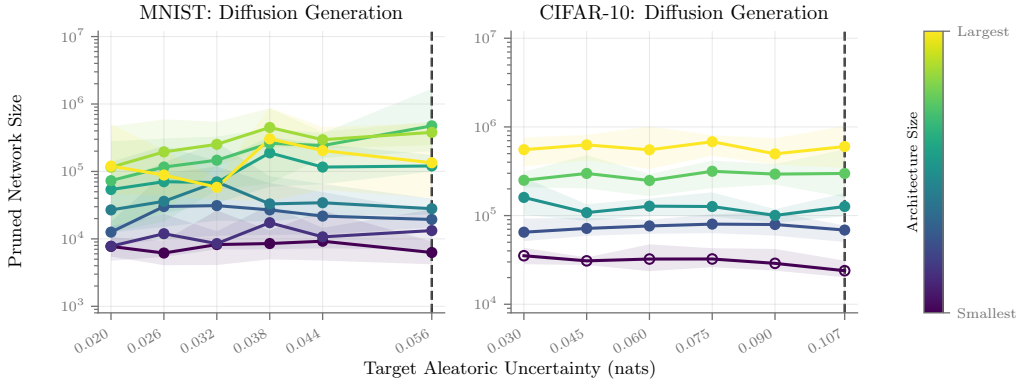

Figure 26: Pruned network size under the standard condition-specific matching criterion. Each architecture size uses the dense test-accuracy population standard deviation computed separately at each target aleatoric uncertainty as its permitted accuracy drop. Pruned network sizes use per-run linear interpolation between adjacent accuracy–parameter-count checkpoints. Curves fix architecture size, and the MNIST and CIFAR-10 panels use independently scaled logarithmic vertical axes. Filled markers indicate that every dense replicate memorizes; open markers indicate that at least one does not. Dashed vertical black lines mark the estimated uncertainty of the original datasets.

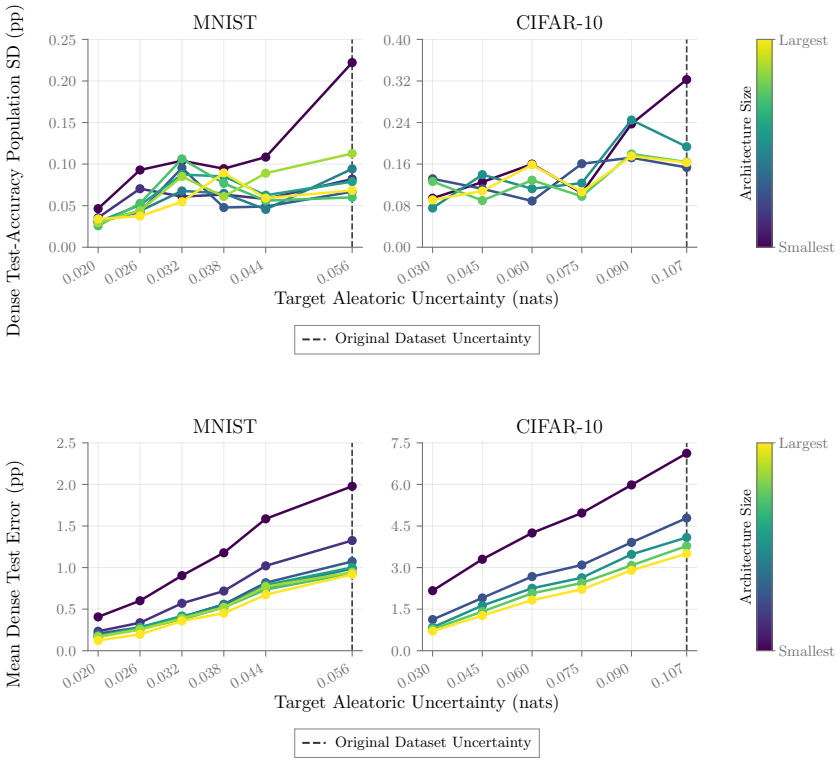

Figure 27: Dense test-accuracy population standard deviation (top) and mean dense test error (bottom), in percentage points, as functions of target aleatoric uncertainty. Curves fix architecture size. Approaching the accuracy ceiling accompanies a contraction in the target-specific tolerance used by the condition-specific criterion.

Table 11: Percentage reduction in mean pruned network size from the original-data target to the lowest target under every fixed absolute accuracy-drop tolerance. Column headings give the tolerance in percentage points; architecture denotes MLP hidden width for MNIST and WRN-28 widen factor for CIFAR-10. Each entry is $100(1 - \bar{M}_{\text{low}}/\bar{M}_{\text{orig}})$, computed from the nine-run means before rounding; negative values indicate larger pruned networks at the lowest target. No endpoint selection in this table is floor-censored.

Dataset	Architecture	0.10	0.25	0.50	1.00	1.50	2.00
MNIST	32	46.5	43.1	40.0	40.2	35.3	31.1
	64	48.8	53.2	55.1	47.7	42.4	42.1
	128	46.6	52.5	55.6	57.4	48.8	40.8
	256	52.0	55.7	58.4	61.2	44.5	35.0
	512	85.3	83.9	80.6	80.6	56.6	33.7
	1024	94.4	92.8	87.1	81.0	44.4	29.2
	2048	94.8	93.1	89.8	76.6	21.2	15.9
	4096	61.2	43.9	42.7	46.9	43.1	32.3
CIFAR-10	0.5	-26.7	-7.0	-0.7	10.8	13.4	15.2
	1	5.3	14.6	23.5	27.5	25.8	24.7
	2	-2.2	31.4	39.1	37.7	38.2	36.9
	4	17.2	23.5	28.1	31.9	33.3	34.2
	8	28.4	37.3	41.6	43.3	44.0	40.3

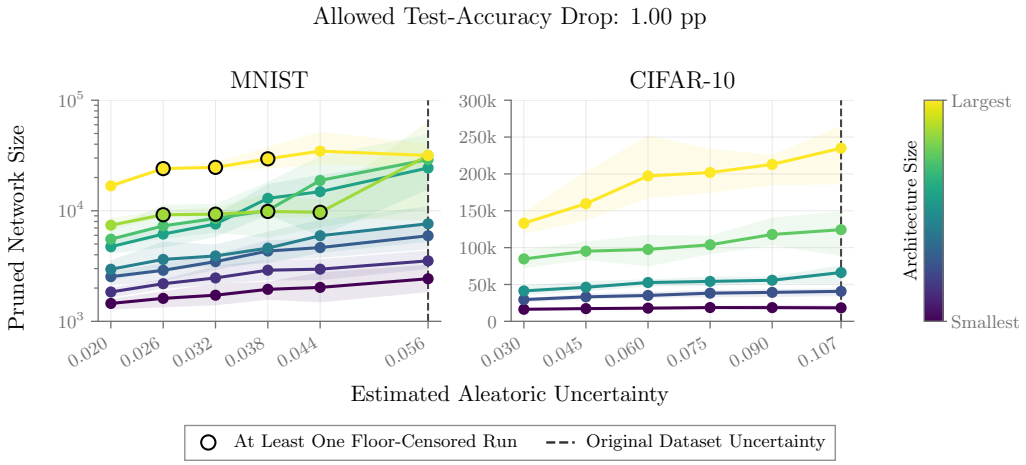

Figure 28: Pruned network size under a fixed absolute accuracy drop of 1.00 percentage point across all six target levels. The same tolerance is used for every architecture size and target aleatoric uncertainty. Lines show nine-run means, shaded bands span their observed minima and maxima, and open markers indicate at least one floor-censored run.

F.7 Sensitivity to Test-Optimal Checkpoint Selection

We apply the test-optimal checkpoint-selection rule in Appendix D.3 to the generated MNIST and CIFAR-10 families. The selected checkpoint maximizes measured test accuracy along each pruning trajectory, including the zero-sparsity checkpoint, with ties resolved by the smallest retained parameter count and then the later pruning operation.

The generated-data trajectories contain both expansion and contraction for 6 of 8 MNIST widths and 3 of 5 CIFAR-10 widen factors, but the endpoint result is not robust to this criterion (Figure 29). At the lowest evaluated aleatoric uncertainty, the selected size is smaller than at the original-data-uncertainty endpoint for only 5 of 13 architecture sizes, compared with 12 of 13 under the primary matching criterion.

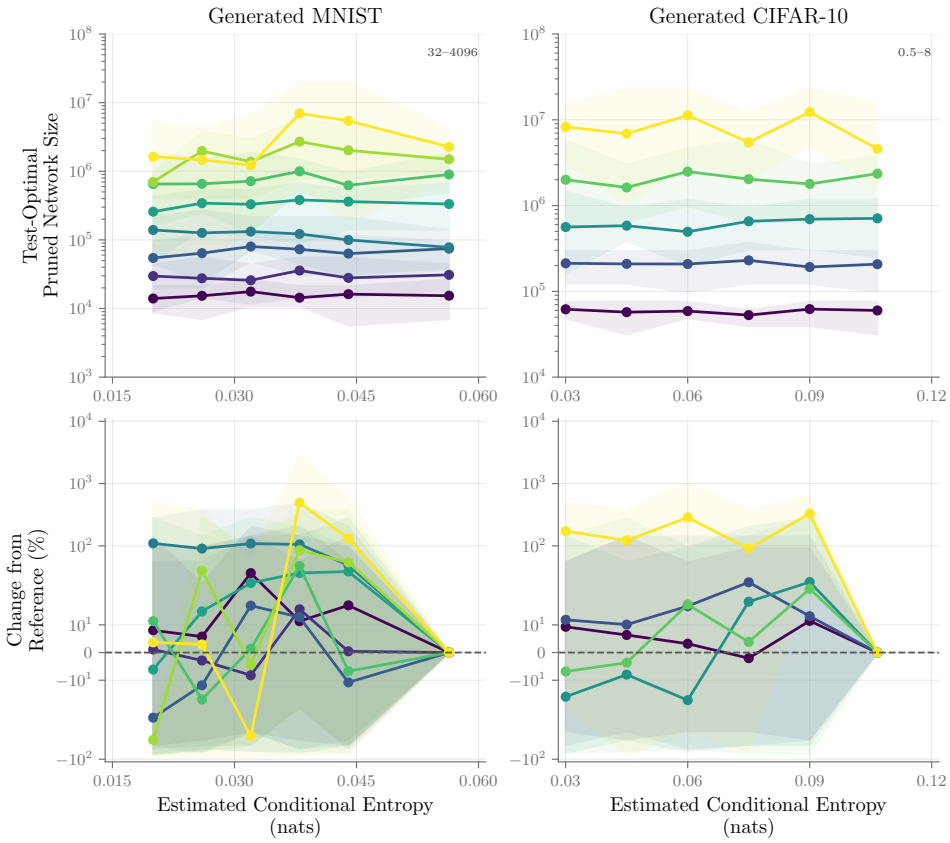

Figure 29: Test-optimal pruned network size for the generated MNIST and CIFAR-10 aleatoric uncertainty families. The top row reports absolute selected counts, and the bottom row reports replicate-level percentage changes from the original-data endpoint. Curves fix architecture size, centers are nine-run means, and bands span the observed run range.

G Extended Related Work

We expand the comparisons most relevant to interpreting pruned network size and our aleatoric uncertainty interventions.

G.1 Pruning as a Measure of Learned Complexity

Pruning-based measures of a learned solution's complexity differ in the performance target they preserve. Kuhn et al. (2021) define prunability as the smallest retained fraction that does not raise training loss and show that it predicts generalization, including around double descent. Ankner et al. (2022) use test-accuracy-matching subnetworks and find that lower intrinsic data dimension generally permits more pruning, whereas extrinsic dimension has a weaker effect and their controlled task-dimensionality intervention has no clear effect. Arora et al. (2023) use the test-optimal checkpoint on an iterative-magnitude-pruning trajectory as an effective parameter count, finding that differently sized parent models can yield best-pruned models of similar size and that this size increases under controlled notions of task difficulty. Our pruned network size instead identifies the smallest checkpoint that matches its own zero-sparsity reference. It can change because either the pruning trajectory or the reference target changes, motivating our separation of uncertainty overhead and target relaxation.

Pruning outcomes also depend on the pruning rule, retraining procedure, architecture, and sparsity structure (Blalock et al., 2020; Hoefler et al., 2021). Training a large model before compression can outperform training a small model with the same final parameter count (Chang et al., 2021), and learning-rate rewinding can recover performance after iterative magnitude pruning (Renda et al., 2020). We therefore interpret our parameter counts relative to the specified training and pruning procedure; Appendix D.2 tests sensitivity to the weight-selection rule.

G.2 Pruning Under Noisy Supervision

Label noise includes annotation errors, deliberate corruption, and stochasticity in the conditional label distribution. These sources need not have the same semantics: a corrupted label may have a latent correction, whereas an ambiguous example may support a non-degenerate conditional label distribution. Networks can learn shared patterns before memorizing atypical or randomly labeled examples (Arpit et al., 2017; Liu et al., 2020), and memorization can localize to a small set of neurons (Maini et al., 2023). These observations motivate possible mechanisms but do not determine the size needed to preserve test performance after pruning.

Several studies connect noisy or difficult examples directly to sparse-model behavior. Hooker et al. (2019) identify examples whose predictions are disproportionately affected by compression and find that these examples overrepresent atypical, ambiguous, low-quality, and potentially mislabeled images. Under injected label noise, Jin et al. (2022) find that pruning at generalization-improving sparsities preferentially raises loss on corrupted examples while preserving a better fit to original examples. Dense models trained without examples misclassified by the corresponding sparse models can match or exceed this generalization improvement. This connects pruning to declining to fit noisy examples, but does not test whether reducing aleatoric uncertainty decreases pruned network size.

Jiang et al. (2024) show that a pruned network that performs similarly to its zero-sparsity counterpart on clean data deteriorates when symmetric label noise is introduced. Their TS^3-Net combines pruning with clean-sample selection, semi-supervised learning, and label refinement to train robust subnetworks at prespecified sparsities. Paul et al. (2022) find that using low-difficulty examples during the short pretraining phase of iterative magnitude pruning produces matching initializations more effectively than using the hardest examples, particularly under random label corruption. Because subsequent mask search and sparse training use the complete dataset, this concerns initialization rather than the pruned network size of a lower-uncertainty distribution. Jin et al. (2026) similarly use likely clean and noisy examples to identify parameters to retain or remove. Our experiments instead vary aleatoric uncertainty and measure the size needed to match a condition-specific reference.

Related work studies different forms of nonmonotonic behavior. Sparse double descent describes deterioration, recovery, and eventual collapse of test performance as sparsity increases under label

noise (He et al., 2022; Quétu & Tartaglione, 2024). Zhou et al. (2023) describe regimes resulting from interactions between training *temperature* and pruning *load*. Our expansion and contraction regimes concern changes in pruned network size as aleatoric uncertainty increases.

G.3 Uncertainty Estimation and Label Auditing

Predictive uncertainty can be decomposed into expected model-conditional entropy and model disagreement, often interpreted as proxies for aleatoric and epistemic uncertainty (Houlsby et al., 2011; Depeweg et al., 2018). Bayesian neural networks approximate a parameter posterior, while deep ensembles use independently trained predictors (Kendall & Gal, 2017; Lakshminarayanan et al., 2017). Independently trained ensemble members are not posterior samples, and entropy and mutual information do not universally isolate aleatoric and epistemic uncertainty (Wimmer et al., 2023). We therefore treat mean ensemble-member entropy as a model-dependent proxy for conditional label entropy.

Repeated annotations provide complementary evidence about label distributions. CIFAR-10H records human-judgment distributions on CIFAR-10 test images, and CIFAR-N records human annotation noise on its training sets (Peterson et al., 2019; Wei et al., 2022). Multi-domain repeated-annotation benchmarks distinguish erroneous labels from ambiguous examples, while soft labels can convey information omitted by a single hard label (Schmarje et al., 2022; Sucholutsky et al., 2023).

Confident Learning uses out-of-sample class probabilities to relate observed and latent clean labels (Northcutt et al., 2021). Area-under-the-margin and dataset-cartography methods use training dynamics to identify mislabeled, ambiguous, hard, or influential examples (Pleiss et al., 2020; Swayamdipta et al., 2020). These methods diagnose data relative to cleaner labeling. We instead manipulate aleatoric uncertainty as an experimental variable and study how the resulting distributions affect training and pruning.

G.4 Overparameterization

Double descent and benign-overfitting results show that interpolation alone does not determine generalization (Belkin et al., 2019; Nakkiran et al., 2020; Bartlett et al., 2020; Frei et al., 2022). Pruning can improve generalization in signal–noise models, but aggressive pruning can also destroy useful feature learning (Yang et al., 2025; Vural & Erdogdu, 2024). Our experiments examine how architecture size changes the relationship between aleatoric uncertainty and pruned network size, with a fixed-distribution, fixed-target control in Appendix D.1.

H Limitations

Experimental Scope. Our natural-data experiments use MNIST and CIFAR-10, with multilayer perceptrons and Wide ResNets, respectively. We vary capacity through width within these architecture families. The results do not establish the same regime boundaries for other architectures, depths, datasets, or modalities, nor a universal degree of overparameterization at which expansion begins.

Aleatoric Uncertainty Estimates and Interventions. We estimate conditional label entropy through mean ensemble-member entropy. This is a model-dependent proxy whose scale depends on the teachers and their calibration, rather than an identified ground-truth conditional entropy. Our experiments summarize distributions by mean aleatoric uncertainty; distributions with the same mean but different shapes may produce different training and pruning behavior. The generated families additionally depend on the generator, agreement filter, finite candidate pool, and histogram materialization procedure. The minimum-KL guarantee applies to ideal class–entropy-bin probabilities, with allocation and substitution details reported in Appendix E.

Pruning and Matching Procedure. Pruned network size measures the outcome of a specified training procedure, pruning rule, and accuracy target. It does not identify the minimum parameter count representable by an architecture family. The clean-label Taylor diagnostic in Appendix D.2

demonstrates sensitivity to the weight-selection rule in one setting; it does not establish whether an alternative procedure eliminates uncertainty overhead generally. Finite pruning horizons, the spacing of measured checkpoints, and the rule for interpolating threshold crossings also limit the resolution of the size estimate. The criterion-sensitivity analyses therefore complement the primary dense-reference matching results.

Optimization and Mechanism. Our findings are conditional on the chosen optimizers, regularization, and training budgets. Dense training emphasizes convergence or interpolation rather than validation-based early stopping, and pruning adds substantial finetuning beyond the zero-sparsity checkpoint. The uniform-label-noise experiments use 100 finetuning epochs per round for MNIST and 60 for CIFAR-10, while CIFAR-10 dense training can last up to 1,000 epochs. Differences in the relative recovery budget may affect pruned network size. The ablations distinguish uncertainty overhead from target relaxation but do not isolate the mechanism producing uncertainty overhead.