Appendices

Appendix A

Optimization Methods

> "*Since the building of all the universe is perfect and is created by the wisdom creator, nothing arises in the universe in which one cannot see the sense of some maximum or minimum.*"
>
> – L. Euler

In this chapter, we give a brief introduction to some of the most basic but important optimization algorithms used in this book. The purpose is only to help the reader apply these algorithms to problems studied in this book, not to gain a deep understanding about these algorithms. Hence, we will not provide a thorough justification for the algorithms introduced, in terms of performance guarantees.

A.1 Steepest Descent

Optimization is concerned with the question of how to find where a function, say $\mathcal{L}(\theta)$, reaches its minimum value. Mathematically, this is stated as a problem:

$$\arg\min_{\theta\in\Theta} \mathcal{L}(\theta), \tag{A.1.1}$$

where Θ represents a domain to which the argument θ is confined. Often (and unless otherwise mentioned, in this chapter) Θ is simply $\mathbb{R}^n$. Without loss of generality, we assume that here the function $\mathcal{L}(\theta)$ is smooth[1].

The efficiency of finding the (global) minima depends on what information we have about the function $\mathcal{L}$. For most optimization problems considered in this book, the dimension of θ, say n, is very large. That makes computing or accessing local information about $\mathcal{L}$ expensive. In particular, since the gradient $\nabla\mathcal{L}$ has n entries, it is often reasonable to compute; however, the Hessian $\nabla^2\mathcal{L}$ has n^2 entries which is often wildly impractical to compute (and the same

[1] In case the function $\mathcal{L}$ is not smooth, we replace its gradient with a so-called *subgradient*.

goes for higher-order derivatives). Hence, it is typical to assume that we have the zeroth-order information, i.e., we are able to evaluate $\mathcal{L}(\theta)$, and the first-order information, i.e., we are able to evaluate $\nabla\mathcal{L}(\theta)$. Optimization theorists may rephrase this as saying we have a "first-order *oracle*." All optimization algorithms that we introduce in this section only use a first-order oracle.[2]

A.1.1 Vanilla Gradient Descent for Smooth Problems

The simplest and most widely used method for optimization is *gradient descent* (GD). It was first introduced by Cauchy in 1847. The idea is very simple: starting from an initial state, we iteratively take small steps such that each step reduces the value of the function $\mathcal{L}(\theta)$.

Suppose that the current state is θ. We want to take a small step, say of distance h, in a direction, indicated by a vector $\boldsymbol{v}$, to reach a new state $\theta + h\boldsymbol{v}$ such that the value of the function decreases:

$$\mathcal{L}(\theta + h\boldsymbol{v}) \leq \mathcal{L}(\theta). \tag{A.1.2}$$

To find such a direction $\boldsymbol{v}$, we can approximate $\mathcal{L}(\theta + h\boldsymbol{v})$ through a Taylor expansion around $h = 0$:

$$\mathcal{L}(\theta + h\boldsymbol{v}) = \mathcal{L}(\theta) + h\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle + o(h), \tag{A.1.3}$$

where the inner product here (and in this chapter) will be the ℓ^2 inner product, i.e., $\langle\boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^\top\boldsymbol{y}$. To find the direction of *steepest descent*, we attempt to minimize this Taylor expansion among unit vectors $\boldsymbol{v}$. If $\nabla\mathcal{L}(\theta) = \mathbf{0}$, then the second term above is 0 regardless of the value of $\boldsymbol{v}$, so we cannot attempt to make progress, i.e., the algorithm has converged. On the other hand, if $\nabla\mathcal{L}(\theta) \neq \mathbf{0}$ then it holds

$$\arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^d \\ \|\boldsymbol{v}\|_2=1}} [\mathcal{L}(\theta) + h\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle] = \arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^d \\ \|\boldsymbol{v}\|_2=1}} \langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle = -\frac{\nabla\mathcal{L}(\theta)}{\|\nabla\mathcal{L}(\theta)\|_2}, \tag{A.1.4}$$

In words, this means that the value of $\mathcal{L}(\theta + h\boldsymbol{v})$ decreases the fastest along the direction $\boldsymbol{v} = -\nabla\mathcal{L}(\theta)/\|\nabla\mathcal{L}(\theta)\|_2$, for small enough h. This leads to the gradient descent method: From the current state θ_k ($k = 0, 1, \ldots$), we take a step of size h in the direction of the negative gradient to reach the next iterate,

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}(\theta_k). \tag{A.1.5}$$

The step size h is also called the *learning rate* in machine learning contexts.

[2] We refer the readers to the book by [WM22] for a more systematic introduction to optimization algorithms in a high-dimensional space, including algorithms assuming higher-order oracles.

Step-Size Selection

The remaining question is what the step size h should be? If we choose h to be too small, the value of the function may decrease very slowly, as shown by the plot in the middle in Figure A.1. If h is too large, the value might not even decrease at all, as shown by the plot on the right in Figure A.1.

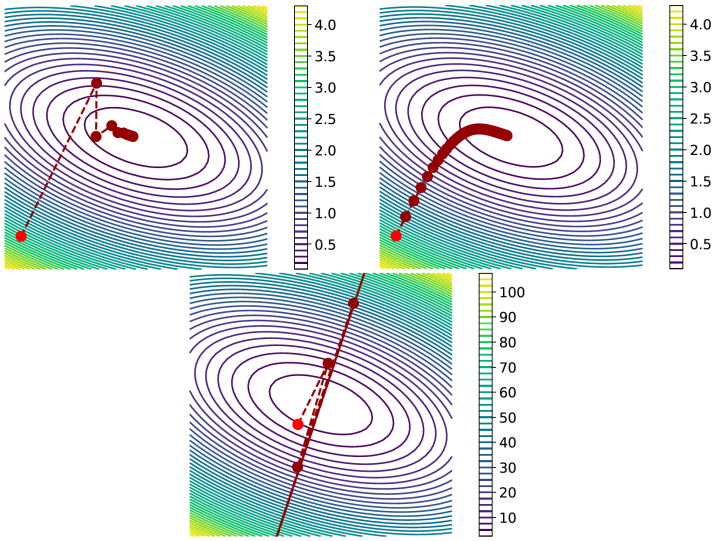

Figure A.1: The effect of the (constant) step size h on the convergence of the gradient descent method.

So the step size h should be chosen based on the landscape of the function $\mathcal{L}(\theta_k)$. Ideally, to choose the best step size h, we can solve the following optimization problem over a single variable h:

$$h = \arg\min_{h' \geq 0} \mathcal{L}(\theta_k - h' \nabla \mathcal{L}(\theta_k)). \tag{A.1.6}$$

This method of choosing the step size is called *line search*; as hinted by the notation, it is usually used to obtain an optimal learning rate *for each iteration* k. However, when the function $\mathcal{L}(\theta_k)$ is complicated, which is usually the case for training a deep neural network, this one-dimensional optimization is very difficult to solve at each iteration of gradient descent.

Then how should we choose a proper step size h? One common and classical approach is to try to obtain a good approximation of the local landscape around the current state θ based on some knowledge about the overall landscape of the function $\mathcal{L}(\theta)$.

Common conditions for the landscape of $\mathcal{L}(\theta)$ include:

- α-Strong Convexity. Recall that $\mathcal{L}$ is α-*strongly convex* if its graph lies

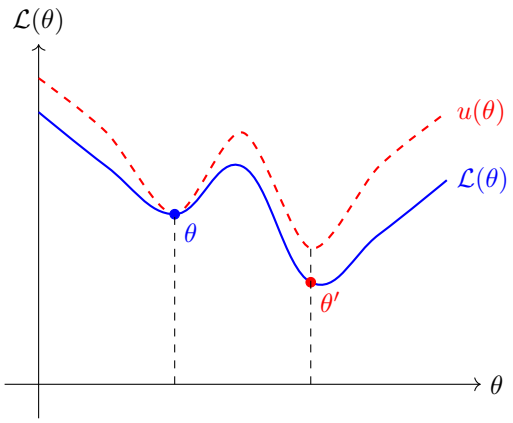

Figure A.2: **Majorization-minimization scheme and intuition.** A function $\mathcal{L}\colon \Theta \to \mathbb{R}$ has a global upper bound $u\colon \Theta \to \mathbb{R}$ which meets $\mathcal{L}$ at at least one point θ. Then, finding the θ' which minimizes u will improve the value of $\mathcal{L}$ from $\mathcal{L}(\theta)$. Note that similar results can be shown about local upper bounds.

above a global quadratic lower bound of slope α, i.e.,

$$\mathcal{L}(\theta') \geq l_{\theta,\alpha}(\theta') \doteq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta \rangle + \frac{\alpha}{2}\|\theta' - \theta\|_2^2 \tag{A.1.7}$$

for any "base point" θ. We say that $\mathcal{L}$ is *convex* if it is 0-strongly convex, i.e., its graph lies above its tangents. It is easy to show (proof as exercise) that strongly convex functions have unique global minima. Another important fact (proof as exercise) is that α-strongly convex twice-differentiable functions $\mathcal{L}$ have (symmetric) Hessians $\nabla^2\mathcal{L}$ whose minimum eigenvalue is $\geq \alpha$. For $\alpha > 0$ this implies the Hessian is symmetric positive definite, and for $\alpha = 0$ (i.e., $\mathcal{L}$ is convex) this implies that the Hessian is symmetric positive semidefinite.

- β-Lipschitz Gradient (also called β-Smoothness). Recall that $\mathcal{L}$ has β-*Lipschitz gradient* if $\nabla\mathcal{L}$ exists and is β-Lipschitz, i.e.,

$$\|\nabla\mathcal{L}(\theta') - \nabla\mathcal{L}(\theta)\|_2 \leq \beta\|\theta' - \theta\|_2. \tag{A.1.8}$$

for any "base point" θ. It is easy to show (proof as exercise) that this is equivalent to $\mathcal{L}$ having a global quadratic upper bound of slope β, i.e.,

$$\mathcal{L}(\theta') \leq u_{\theta,\beta}(\theta') \doteq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \theta' - \theta \rangle + \frac{\beta}{2}\|\theta' - \theta\|_2^2. \tag{A.1.9}$$

for any "base point" θ. Another important fact (proof as exercise) is that convex β-Lipschitz gradient twice-differentiable functions have (symmetric) Hessians $\nabla^2\mathcal{L}$ whose largest eigenvalue is $\leq \beta$.

First, let us suppose that $\mathcal{L}$ has β-Lipschitz gradient (but is not necessarily even convex). We will use this occasion to introduce a common optimization theme: *to minimize $\mathcal{L}$, we can minimize an upper bound on $\mathcal{L}$*, which is justified by the following lemma visualized in Figure A.2.

Lemma A.1 (Majorization-Minimization)**.** *Suppose that $u\colon \Theta \to \mathbb{R}$ is a global upper bound on $\mathcal{L}$, namely $\mathcal{L}(\theta) \le u(\theta)$ for all $\theta \in \Theta$. Suppose that they meet with equality at θ, i.e., $\mathcal{L}(\theta) = u(\theta)$. Then*

$$\theta^+ \in \arg\min_{\theta' \in \Theta} u(\theta') \implies \mathcal{L}(\theta^+) \le u(\theta^+) \le u(\theta) = \mathcal{L}(\theta). \tag{A.1.10}$$

We will use this lemma to show that we can use the Lipschitz gradient property to ensure that each gradient step cannot worsen the value of $\mathcal{L}$. Indeed, at every base point θ, we have that $u_{\theta,\beta}$ is a global upper bound on $\mathcal{L}$, and $u_{\theta,\beta}(\theta) = \mathcal{L}(\theta)$. Hence by Lemma A.1

$$\text{if } \theta^+ \text{ minimizes } u_{\theta,\beta} \text{ then} \quad \mathcal{L}(\theta^+) \le u_{\theta,\beta}(\theta^+) \le u_{\theta,\beta}(\theta) = \mathcal{L}(\theta). \tag{A.1.11}$$

This motivates us, when finding an update to obtain θ_{k+1} from θ_k, we can instead minimize the upper bound $u_{\theta_k,\beta}$ over θ and set that to be θ_{k+1}. By minimizing $u_{\theta_k,\beta}$ (proof as exercise) we get

$$\theta_{k+1} = \theta_k - \frac{1}{\beta}\nabla\mathcal{L}(\theta_k) \implies \mathcal{L}(\theta_{k+1}) \le \mathcal{L}(\theta_k). \tag{A.1.12}$$

This implies that a step size $h = 1/\beta$ is a usable learning rate, but it does not provide a convergence rate or certify that $\mathcal{L}(\theta_k)$ actually converges to $\min_\theta \mathcal{L}(\theta)$. This requires a little more rigor, which we now pursue.

Now, let us suppose that $\mathcal{L}$ is α-strongly convex, has β-Lipschitz gradient, and has global optimum $\theta^\star$. We will show that θ_k will converge directly to the unique global optimum $\theta^\star$, which is a very strong form of convergence. In particular, we will bound $\|\theta^\star - \theta_k\|_2$ using both strong convexity and Lipschitzness of the gradient of $\mathcal{L}$, i.e., taking a look at the neighborhood around θ_k:[3]

$$\begin{aligned}
\|\theta^\star - \theta_{k+1}\|_2^2 &\le \|\theta^\star - \theta_k + h\nabla\mathcal{L}(\theta_k)\|_2^2 && \text{(A.1.13)}\\
&= \|\theta^\star - \theta_k\|_2^2 + 2h\langle\nabla\mathcal{L}(\theta_k), \theta^\star - \theta_k\rangle + h^2\|\nabla\mathcal{L}(\theta_k)\|_2^2 && \text{(A.1.14)}\\
&\le \|\theta^\star - \theta_k\|_2^2 + 2h\left(\mathcal{L}(\theta^\star) - \mathcal{L}(\theta_k) - \frac{\alpha}{2}\|\theta^\star - \theta_k\|_2^2\right) + h^2\|\nabla\mathcal{L}(\theta_k)\|_2^2 && (\alpha\text{-SC})\\
& && \text{(A.1.15)}\\
&= (1 - \alpha h)\,\|\theta^\star - \theta_k\|_2^2 + 2h(\mathcal{L}(\theta^\star) - \mathcal{L}(\theta_k)) + h^2\|\nabla\mathcal{L}(\theta_k)\|_2^2 && \\
& && \text{(A.1.16)}\\
&\le (1 - \alpha h)\,\|\theta^\star - \theta_k\|_2^2 + 2h(\mathcal{L}(\theta^\star) - \mathcal{L}(\theta_k)) + 2h^2\beta(\mathcal{L}(\theta_k) - \mathcal{L}(\theta^\star)) && (\beta\text{-LG})\\
& && \text{(A.1.17)}
\end{aligned}$$

[3] In this proof the β-Lipschitz Gradient invocation step is a little non-trivial. We also leave this step as an exercise, with the hint to plug in $\theta = \theta_0 - h\nabla\mathcal{L}(\theta_0)$ into the Lipschitz gradient identity.

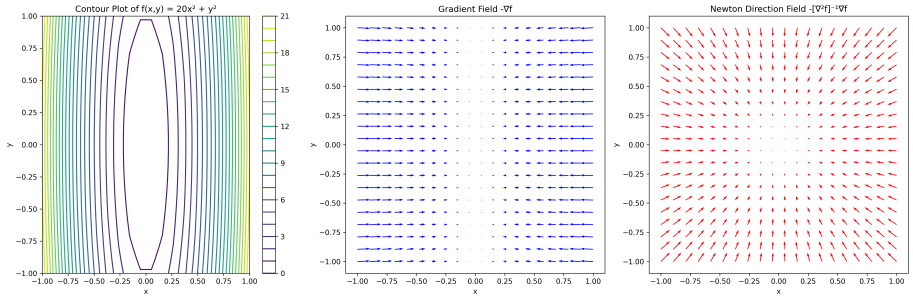

Figure A.3: **The negative gradient $-\nabla\mathcal{L}_\lambda$ and pre-conditioned (Newton's method step) vector field** $-[\nabla^2\mathcal{L}_\lambda]^{-1}[\nabla\mathcal{L}_\lambda]$ where $\lambda = 19$. There is a section of the space where following the negative gradient vector field makes very little progress towards finding the minimum, but in all cases following the Newton's method vector field achieves equal speed of progress towards the optimum since the gradient is whitened. Since the Hessian here is diagonal, adaptive learning rate algorithms (e.g. Adam, as will be discussed later in the section) can make similar progress as Newton's method, but a non-axis-aligned Hessian may even prevent Adam from succeeding quickly.

$$= (1 - \alpha h)\,\|\theta^\star - \theta_k\|_2^2 - 2h(1 - \beta h)(\mathcal{L}(\theta_k) - \mathcal{L}(\theta^\star)). \tag{A.1.18}$$

In order to ensure that the gradient descent iteration makes progress we must pick the step size so that $1 - \beta h \geq 0$, i.e., $h \leq 1/\beta$. If such a setting occurs, then

$$\|\theta^\star - \theta_{k+1}\|_2^2 \leq (1 - \alpha h)\|\theta^\star - \theta_k\|_2^2 \leq (1 - \alpha h)^2\|\theta^\star - \theta_{k-1}\|_2^2 \leq \cdots \tag{A.1.19}$$

$$\leq (1 - \alpha h)^{k+1}\|\theta^\star - \theta_0\|_2^2. \tag{A.1.20}$$

In order to minimize the right-hand side, we can set $h = 1/\beta$, which obtains

$$\|\theta^\star - \theta_{k+1}\|_2^2 \leq (1 - \alpha/\beta)^{k+1}\|\theta^\star - \theta_0\|_2^2, \tag{A.1.21}$$

showing convergence to global optimum with exponentially decaying error. Notice that here we used a convergence rate to obtain a favorable *step size* of $h = 1/\beta$. This motif will re-occur in this section.

We end this section with a caveat: learning a global optimum is (usually) impractically hard. Under certain conditions, we can ensure that the gradient descent iterates converge to a *local optimum*. Also, under more relaxed conditions, we can ensure *local* convergence, i.e., that the iterates converge to a (global or local) optimum if the sequence is initialized close enough to the optimum.

A.1.2 Preconditioned Gradient Descent for Badly-Conditioned Problems

Newton's Method

There are some smooth problems and strongly convex problems on which gradient descent nonetheless does quite poorly. For example, let $\lambda \geq 0$ and let $\mathcal{L}_\lambda \colon \mathbb{R}^2 \to \mathbb{R}$ of the form

$$\mathcal{L}_\lambda(\theta) = \mathcal{L}_\lambda\left(\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}\right) \doteq \frac{1}{2}\left\{(1+\lambda)\theta_1^2 + \theta_2^2\right\} = \frac{1}{2}\theta^\top \begin{bmatrix}1+\lambda & 0 \\ 0 & 1\end{bmatrix}\theta. \tag{A.1.22}$$

This problem is 1-strongly convex and has $(1+\lambda)$-Lipschitz gradient. The convergence rate is then geometric with rate $1 - 1/(1+\lambda)$. For large λ, this is still not very fast. In this section, we will introduce a class of optimization problems which can successfully optimize such badly-conditioned functions.

The key lies in the objective's *curvature*, which is given by the Hessian. Suppose that (as a counterfactual) we had a *second-order* oracle which would allow us to compute $\mathcal{L}(\theta)$, $\nabla\mathcal{L}(\theta)$, and $\nabla^2\mathcal{L}(\theta)$. Then, instead of picking a descent direction $\boldsymbol{v}$ to optimize the first-order Taylor expansion around θ, we could optimize the second-order Taylor expansion instead. Intuitively this would allow us to incorporate curvature information into the update.

Let us carry out this computation. The second-order Taylor expansion of $\mathcal{L}(\theta + h\boldsymbol{v})$ around $h = 0$ is

$$\mathcal{L}(\theta + h\boldsymbol{v}) = \mathcal{L}(\theta) + h\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle + \frac{1}{2}h^2\langle[\nabla^2\mathcal{L}(\theta)]\boldsymbol{v}, \boldsymbol{v}\rangle + o(h^2). \tag{A.1.23}$$

Then we can compute the descent direction:

$$\arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \|\boldsymbol{v}\|_2=1}} \left[\mathcal{L}(\theta) + h\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle + \frac{1}{2}h^2\langle[\nabla^2\mathcal{L}(\theta)]\boldsymbol{v}, \boldsymbol{v}\rangle\right] \tag{A.1.24}$$

$$= \arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \|\boldsymbol{v}\|_2=1}} \left[\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle + \frac{1}{2}h\langle[\nabla^2\mathcal{L}(\theta)]\boldsymbol{v}, \boldsymbol{v}\rangle\right]. \tag{A.1.25}$$

This optimization problem is a little difficult to solve because of the constraint $\|\boldsymbol{v}\|_2 = 1$. But in practice we never normalize the descent direction $\boldsymbol{v}$ and use the step size h to control the size of the update. So let us just solve the above problem over all vectors $\boldsymbol{v} \in \mathbb{R}^n$:[4]

$$\arg\min_{\boldsymbol{v}\in\mathbb{R}^n} \left[\langle\nabla\mathcal{L}(\theta), \boldsymbol{v}\rangle + \frac{1}{2}h\langle[\nabla^2\mathcal{L}(\theta)]\boldsymbol{v}, \boldsymbol{v}\rangle\right] = -\frac{1}{h}[\nabla^2\mathcal{L}(\theta)]^{-1}[\nabla\mathcal{L}(\theta)]. \tag{A.1.26}$$

We can thus use the steepest descent iteration

$$\theta_{k+1} = \theta_k - [\nabla^2\mathcal{L}(\theta_k)]^{-1}[\nabla\mathcal{L}(\theta_k)], \tag{A.1.27}$$

[4] If $\nabla^2\mathcal{L}(\theta)$ is not invertible, then we can replace $[\nabla^2\mathcal{L}(\theta)]^{-1}$ with the Moore-Penrose pseudoinverse of $\nabla^2\mathcal{L}(\theta)$.

(this is the celebrated *Newton's method*), or

$$\theta_{k+1} = \theta_k - h[\nabla^2 \mathcal{L}(\theta_k)]^{-1}[\nabla \mathcal{L}(\theta_k)], \tag{A.1.28}$$

(which is called *underdamped Newton's method*). Since the second-order quadratic $\mathcal{L}_\lambda$ is equal to its second-order Taylor expansion, if we run Newton's method for *one step*, we will achieve the global minimum in *one step* no matter how large λ is. Figure A.3 gives some intuition about poorly conditioned functions and the gradient steps versus Newton's steps.

PGD

In practice, we do *not* have a second-order oracle which allows us to compute $\nabla^2 \mathcal{L}(\theta)$. Instead, we can attempt to *learn an approximation to it* alongside the parameter update θ_{k+1} from θ_k.

How do we learn an approximation to it? We shall find some equations which the Hessian's inverse satisfies and then try to update our approximation so that it satisfies the equations. Namely, taking the Taylor series of $\nabla \mathcal{L}(\theta + \delta_\theta)$ around point θ, we obtain

$$\underbrace{\nabla \mathcal{L}(\theta + \delta_\theta) - \nabla \mathcal{L}(\theta)}_{\doteq \delta_{\boldsymbol{g}}} = [\nabla^2 \mathcal{L}(\theta)]\delta_\theta + o(\|\delta_\theta\|_2). \tag{A.1.29}$$

In this case we have

$$\delta_{\boldsymbol{g}} \approx [\nabla^2 \mathcal{L}(\theta)]\delta_\theta \implies \delta_\theta \approx [\nabla^2 \mathcal{L}(\theta)]^{-1}\delta_{\boldsymbol{g}} \tag{A.1.30}$$

We can now try to learn a symmetric positive semidefinite pre-conditioner $P \in \mathbb{R}^{n \times n}$ such that

$$\delta_\theta \approx P\delta_{\boldsymbol{g}}, \tag{A.1.31}$$

updating it at each iteration along with θ_k. Namely, we have the *PSGD* iteration

$$P_k = \text{PreconditionerUpdate}(P_{k-1}; \theta_k, \nabla \mathcal{L}(\theta_k)) \tag{A.1.32}$$

$$\theta_{k+1} = \theta_k - hP_k \nabla \mathcal{L}(\theta_k). \tag{A.1.33}$$

This update has two problems: how can we even use P (since we already said we cannot store an $n \times n$ matrix) and how can we *update* P at each iteration? The answers are very related; we can never materialize P in computer memory, but we can represent it using a low-rank factorization (or comparable methods such as *Kronecker factorization* which is particularly suited to the form of deep neural networks). Then the preconditioner update step is designed to exploit the structure of the preconditioner representation.

We end this subsection with a caveat: in deep learning, for example, $\mathcal{L}$ is not a convex function and so Newton's method (and approximations to it) do not make sense. In this case we look at the geometric intuition of Newton's method on convex functions, say from Figure A.3: the inverse Hessian *whitens* the gradients. Thus instead of a Hessian-approximating preconditioner, we can adjust

the above procedures to learn a more general whitening transformation for the gradient. This is the idea behind the original proposal of PSGD [Li17], which contains more information about how to store and update the preconditioner, and more modern optimizers like Muon [LSY+25].

A.1.3 Proximal Gradient Descent for Non-Smooth Problems

Even in very toy problems, however, such as LASSO or dictionary learning, the problem is not strongly convex but rather just convex, and the objective is no longer just smooth but rather the sum of a smooth function and a non-smooth regularizer (such as the ℓ^1 norm). Such problems are solved by *proximal optimization algorithms*, which generalize steepest descent to non-smooth objectives.

Formally, let us say that

$$\mathcal{L}(\theta) \doteq \mathcal{S}(\theta) + \mathcal{R}(\theta) \tag{A.1.34}$$

where $\mathcal{S}$ is smooth, say with β-Lipschitz gradient, and $\mathcal{R}$ is non-smooth (i.e., rough). The proximal gradient algorithm generalizes the steepest descent algorithm, by using the majorization-minimization framework (i.e., Lemma A.1) with a different global upper bound. Namely, we construct such an upper bound by asking: what if we take the Lipschitz gradient upper bound of S but *leave* R *alone*? Namely, we have

$$\mathcal{L}(\theta') = \mathcal{S}(\theta') + \mathcal{R}(\theta') \leq u_{\theta,\beta}(\theta') \doteq \mathcal{S}(\theta) + \langle \nabla \mathcal{S}(\theta), \theta' - \theta \rangle + \frac{\beta}{2}\|\theta' - \theta\|_2^2 + \mathcal{R}(\theta'). \tag{A.1.35}$$

Note that (proof as exercise)

$$\arg\min_{\theta'} u_{\theta,\beta}(\theta') = \arg\min_{\theta'} \left[\frac{\beta}{2} \left\| \theta' - \left(\theta - \frac{1}{\beta} \nabla \mathcal{S}(\theta) \right) \right\|_2^2 + \mathcal{R}(\theta') \right]. \tag{A.1.36}$$

Now if we try to minimize the upper bound $u_{\theta,\beta}$, we are picking a θ' that:

- is close to the gradient update $\theta - \frac{1}{\beta}\nabla \mathcal{S}(\theta)$;
- has a small value of the regularizer $\mathcal{R}(\theta')$

and trades off these properties according to the smoothness parameter β. Accordingly, let us define the proximal operator

$$\mathrm{prox}_{h,\mathcal{R}}(\theta) \doteq \arg\min_{\theta'} \left[\frac{1}{2h}\|\theta' - \theta\|_2^2 + \mathcal{R}(\theta') \right]. \tag{A.1.37}$$

Then, we can define the *proximal gradient descent* iteration which, at each iteration, minimizes the upper bound $u_{\theta_k,h^{-1}}$, i.e.,

$$\theta_{k+1} = \mathrm{prox}_{h,\mathcal{R}}(\theta_k - h\nabla \mathcal{S}(\theta_k)). \tag{A.1.38}$$

Convergence proofs are possible when $h \leq 1/\beta$, but we do not give any in this section.

One remaining question is: how can we compute the proximal operator? At first glance, it seems like we have traded one intractable minimization problem for another. Since we have not made any assumption on $\mathcal{R}$ so far, the framework works even when $\mathcal{R}$ is a very complex function (such as a neural network loss), which would require us to solve a neural network training problem in order to compute a single proximal operator. However, in practice, for simple regularizers $\mathcal{R}$ such as those we use in this manuscript, there exist proximal operators which are easy to compute or even in closed-form. We give a few below (the proofs are an exercise).

Example A.1. Let $\Gamma \subseteq \Theta$ be a set, and let χ_Γ be the characteristic function on Γ, i.e.,

$$\chi_\Gamma(\theta) \doteq \begin{cases} 0, & \text{if } \theta \in \Gamma \\ +\infty, & \text{if } \theta \notin \Gamma. \end{cases} \tag{A.1.39}$$

Then the proximal operator of χ_Γ is a projection, i.e.,

$$\operatorname{prox}_{h,\chi_\Gamma}(\theta) = \arg\min_{\theta' \in \Gamma} \frac{1}{2}\|\theta' - \theta\|_2^2 = \arg\min_{\theta' \in \Gamma} \|\theta' - \theta\|_2. \tag{A.1.40}$$

■

Example A.2. The ℓ^1 norm has a proximal operator which performs soft thresholding:

$$S_h(\theta) \doteq \operatorname{prox}_{h,\lambda\|\cdot\|_1}(\theta) = \arg\min_{\theta'} \left[\frac{1}{2h}\|\theta' - \theta\|_2^2 + \lambda\|\theta'\|_1\right] \tag{A.1.41}$$

then $S_h(\theta)$ is defined coordinate-wise by

$$S_h(\theta)_i = \begin{cases} \theta_i - h\lambda, & \text{if } \theta_i \geq h\lambda \\ 0, & \text{if } \theta_i \in [-h\lambda, h\lambda] \\ \theta_i + h\lambda, & \text{if } \theta_i \leq -h\lambda \end{cases} = \begin{cases} \max\{|\theta_i| - h\lambda, 0\}\operatorname{sign}(\theta_i), & \text{if } |\theta_i| \geq h\lambda \\ 0, & \text{if } |\theta_i| < h\lambda. \end{cases} \tag{A.1.42}$$

The proximal gradient operation with the smooth part of the objective being least-squares and the non-smooth part being the ℓ^1 norm (hence using this soft thresholding proximal operator) is called the Iterative Shrinkage-Thresholding Algorithm (ISTA). ■

Example A.3. In Chapter 5 we use a proximal operator corresponding to the ℓ^1 norm plus the characteristic function for the positive orthant $\mathbb{R}^n_+ \doteq \{\boldsymbol{x} \in \mathbb{R}^n : x_i \geq 0 \ \forall i\}$, namely

$$T_h(\theta) \doteq \operatorname{prox}_{h,\lambda\|\cdot\|_1 + \chi_{\mathbb{R}^n_+}}(\theta) = \arg\min_{\theta' \in \mathbb{R}^n_+} \left[\frac{1}{2h}\|\theta' - \theta\|_2^2 + \lambda\|\theta'\|_1\right], \tag{A.1.43}$$

then T_h is defined as

$$T_h(\theta)_i \doteq \max\{\theta_i - h\lambda, 0\}. \tag{A.1.44}$$

This proximal operator yields the non-negative ISTA that is invoked in Chapter 5 and beyond. ■

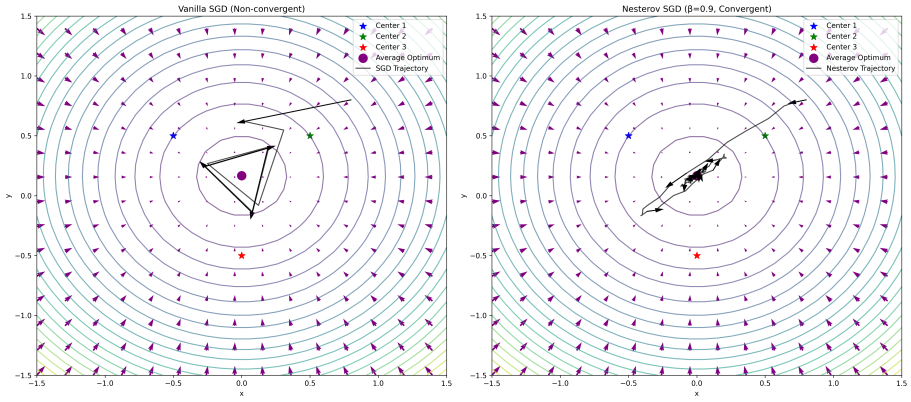

Figure A.4: **Stochastic gradient descent may not converge, even for very benign objectives, but momentum SGD converges.** For even simple quadratic objectives, stochastic gradient descent iterates may pinball around the global optimum, whereas momentum-averaged gradients align to point to the optimal value.

A.1.4 Stochastic Gradient Descent for Large-Scale Problems

In deep learning, the objective function $\mathcal{L}$ usually cannot be computed exactly, and instead at each optimization step it is *estimated* using finite samples (say, using a mini-batch). A common way to model this situation is to define a *stochastic loss function* $\mathcal{L}_\omega(\theta)$ where ω is some "source of randomness". For example, ω could contain the indices of the samples in a batch over which to compute the loss. Then, we would like to minimize $\mathcal{L}(\theta) \doteq \mathbb{E}_\omega[\mathcal{L}_\omega(\theta)]$ over θ, given access to a *stochastic first-order oracle*: given θ, we can sample ω and compute $\mathcal{L}_\omega(\theta)$ and $\nabla_\theta \mathcal{L}_\omega(\theta)$. This minimization problem is called a *stochastic optimization problem.*

The basic first-order stochastic algorithm is *stochastic gradient descent*: at each iteration k we sample ω_k, define $\mathcal{L}_k \doteq \mathcal{L}_{\omega_k}$, and perform a gradient step on $\mathcal{L}_k$, i.e.,

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}_k(\theta_k). \tag{A.1.45}$$

However, even for very simple problems we cannot expect the same type of convergence as we obtained in gradient descent. For example, suppose that there are m possible values for $\omega \in \{1, \dots, m\}$ which it takes with equal probability, and there are m possible targets $\xi_1, \dots, \xi_m$, such that the loss function $\mathcal{L}_\omega$ is

$$\mathcal{L}_\omega(\theta) \doteq \frac{1}{2}\|\theta - \xi_\omega\|_2^2. \tag{A.1.46}$$

Then $\arg\min_\theta \mathbb{E}[\mathcal{L}_\omega(\theta)] = \frac{1}{m}\sum_{i=1}^m \xi_i$, but stochastic gradient descent can "pinball" around the global optimum value, and not converge, as visualized in Figure A.4.

In order to fix this, we can either average the parameters θ_k or average the gradients $\nabla\mathcal{L}_k(\theta_k)$ over time. If we average the parameters θ_k, then (using Figure A.4 as a mental model) the issue of pinballing is straightforwardly not possible, since the average iterate will grow closer to the center. As such, most theoretical convergence proofs consider the convergence of the average iterate $\frac{1}{k}\sum_{i=0}^{k}\theta_i$ to the global minimum. If we average the gradients, we will eventually learn an average gradient $\frac{1}{k}\sum_{i=0}^{k}\nabla\mathcal{L}_k(\theta_k)$ which does not change much at each iteration and therefore does not pinball.

In practice, instead of using an arithmetic average, we take an *exponentially moving average* (EMA) of the parameters (this is called *Polyak averaging*) or of the gradients (this is called *momentum*). Momentum is more popular and we will study it here.

A stochastic gradient descent iteration with momentum is as follows:

$$\boldsymbol{g}_k = \beta\boldsymbol{g}_{k-1} + (1-\beta)\nabla\mathcal{L}_k(\theta_k) \tag{A.1.47}$$

$$\theta_{k+1} = \theta_k - h\boldsymbol{g}_k. \tag{A.1.48}$$

We do not go through a convergence proof (see Chapter 7 of [GG23] for an example). However, momentum handles our toy case in Figure A.4 easily (see the right-hand figure): it stops pinballing and eventually converges to the global optimum.

We end with a caveat: one can show that Polyak momentum and Nesterov momentum are equivalent, for certain choices of parameter settings. Then it is also possible to show that a decaying learning rate schedule (i.e., the learning rate h depends on the iteration k, and its limit is $h_k \to 0$ as $k \to \infty$) with plain SGD (or PSGD) can mimic the effect of momentum. Namely, [DCM+23] shows that if the SGD algorithm lasts K iterations, the gradient norms are bounded $\|\nabla\mathcal{L}(\theta_k)\|_2 \leq G$, and we define $D \doteq \|\theta_0 - \theta^\star\|_2$, then plain SGD iterates θ_k satisfy the rate $\mathbb{E}[\mathcal{L}(\theta_k) - \mathcal{L}(\theta^\star)] \leq DG/\sqrt{K}$ — but only so long as the learning rate $h_k = (D/[G\sqrt{K}])(1-k/K)$ decays *linearly* with time. This matches learning rate schedules used in practice. Indeed, surprisingly, such a theory of convex optimization can predict many empirical phenomena in deep networks [SHT+25], despite deep learning optimization being highly non-convex and non-smooth in the worst case. It is so far unclear why this is the case.

A.1.5 Putting Everything Together: Adam

The gradient descent scheme proposes an iteration of the form

$$\theta_{k+1} = \theta_k + h\boldsymbol{v}_k, \tag{A.1.49}$$

where (recall) $\boldsymbol{v}_k$ is chosen to be (proportional to) the steepest descent vector in the Euclidean norm:

$$\boldsymbol{v}_k = -\frac{\nabla\mathcal{L}(\theta_k)}{\|\nabla\mathcal{L}(\theta_k)\|_2} \in \arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \|\boldsymbol{v}\|_2=1}} \langle\nabla\mathcal{L}(\theta_k), \boldsymbol{v}\rangle. \tag{A.1.50}$$

However, in the context of deep learning optimization, there is absolutely nothing which implies that we have to use the Euclidean norm; indeed the "natural geometry" of the space of parameters is not well-respected by the Euclidean norm, since small changes in the parameter space can lead to very large differences in the output space, for a particular fixed input to the network. If we were instead to use a generic norm $\|\cdot\|$ on the parameter space $\mathbb{R}^n$, we would get some other quantity corresponding to the so-called *dual norm*:

$$\boldsymbol{v}_k \in \arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \|\boldsymbol{v}\|=1}} \langle \nabla\mathcal{L}(\theta_k), \boldsymbol{v}\rangle. \tag{A.1.51}$$

For instance, if we were to use the ℓ^∞-norm, it is possible to show that

$$\boldsymbol{v}_k = -\operatorname{sign}(\nabla\mathcal{L}(\theta_k)) \in \arg\min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \|\boldsymbol{v}\|_\infty=1}} \langle \nabla\mathcal{L}(\theta_k), \boldsymbol{v}\rangle, \tag{A.1.52}$$

where $\operatorname{sign}(\boldsymbol{x})_i = \operatorname{sign}(x_i) \in \{-1, 0, 1\}$. Thus if we were so-inclined, we could use the so-called *sign-gradient descent*:

$$\theta_{k+1} = \theta_k - h\operatorname{sign}(\nabla\mathcal{L}(\theta_k)). \tag{A.1.53}$$

From sign-gradient descent, we can derive the famous Adam optimization algorithm. Note that for a scalar $x \in \mathbb{R}$ we can write

$$\operatorname{sign}(x) = \frac{x}{|x|} = \frac{x}{\sqrt{x^2}}. \tag{A.1.54}$$

Similarly, for a vector $\boldsymbol{x} \in \mathbb{R}^n$ we write (where $\odot$ and $\oslash$ are element-wise multiplication and division)

$$\operatorname{sign}(\boldsymbol{x}) = \boldsymbol{x} \oslash [\boldsymbol{x}^{\odot 2}]^{\odot(1/2)}. \tag{A.1.55}$$

Using this representation we can write (A.1.53) as

$$\theta_{k+1} = \theta_k - h([\nabla\mathcal{L}(\theta_k)] \oslash [\nabla\mathcal{L}(\theta_k)^{\odot 2}]^{\odot\frac{1}{2}}). \tag{A.1.56}$$

Now consider the stochastic regime where we are optimizing a different loss $\mathcal{L}_k$ at each iteration. In SGD, we "tracked" (i.e., took an average of) the gradients using momentum. Here, we can track both the gradient and the squared gradient using momentum, i.e.,

$$\boldsymbol{g}_k = \beta^1\boldsymbol{g}_{k-1} + (1-\beta^1)\nabla\mathcal{L}_k(\theta_k) \tag{A.1.57}$$

$$\boldsymbol{s}_k = \beta^2\boldsymbol{s}_{k-1} + (1-\beta^2)[\nabla\mathcal{L}_k(\theta_k)]^{\odot 2} \tag{A.1.58}$$

$$\theta_{k+1} = \theta_k - h\boldsymbol{g}_k \oslash \boldsymbol{s}_k^{\odot\frac{1}{2}}, \tag{A.1.59}$$

where $\beta^i \in [0, 1]$ are the momentum parameters. The algorithm presented by this iteration is the celebrated *Adam* optimizer,[5] which is the most-used optimizer in deep learning. While convergence proofs of Adam are more involved, it

[5] In order to avoid division-by-zero errors, we divide by $\boldsymbol{s}_k^{\odot(1/2)} + \varepsilon\mathbf{1}_n$ where ε is small, say on the order of 10^{-8}.

falls out of the same steepest descent principle we used so far, and so we should expect that given a small enough learning rate, each update should improve the loss.

Another way to view Adam, which partially explains its empirical success, is that it dynamically updates the learning rates for each parameter based on the squared gradients. In particular, notice that we can write

$$\theta_{k+1} = \theta_k - \eta_k \odot \boldsymbol{g}_k \qquad \text{where} \qquad \eta_k = h\boldsymbol{s}_k^{\odot(-\frac{1}{2})} \tag{A.1.60}$$

where η_k is the parameter-wise adaptively set learning rate. This scheme is called adaptive because if the gradient of a particular parameter is large up to iteration k, then the learning rate for this parameter becomes smaller to compensate, and vice versa, as can be seen from the above equation.

A.2 Computing Gradients via Automatic Differentiation

Above, we discussed several optimization algorithms for deep networks which assumed access to a *first-order oracle*, i.e., a device which would allow us to compute $\mathcal{L}(\theta)$ and $\nabla\mathcal{L}(\theta)$. For simple functions $\mathcal{L}$, it is possible to do this by hand. However, for deep neural networks, this quickly becomes tedious, and hinders rapid experimentation. Hence we require a general algorithm which would allow us to efficiently compute the gradients of arbitrary (sub)differentiable network architectures.

In this section, we introduce the basics of *automatic differentiation* (AD or *autodiff*), which is a computationally efficient way to compute gradients and Jacobians of general functions $f : \mathbb{R}^m \to \mathbb{R}^n$. We will show how this leads to the backpropagation algorithm for computing gradients of loss functions involving neural networks. A summary of the structure of this section is as follows:

1. We introduce **differentials**, a convenient formalism for calculating and organizing the derivatives of functions between high-dimensional parameter spaces (which may themselves be products of other spaces involving matrices, tensors, etc.).

2. We describe the basics of **forward-mode** and **reverse-mode** automatic differentiation, which involves considerations that are important for efficient computation of gradients/Jacobians for different kinds of functions arising in machine learning applications.

3. We describe **backpropagation** in the special case of a loss function applied to a stack-of-layers neural network as an instantiation of reverse-mode automatic differentiation.

Our treatment will err on the mathematical side, to give the reader a deep understanding of the underlying mathematics. The reader should ensure to

couple this understanding with a strong grasp of practical aspects of automatic differentiation for deep learning, for example as manifested in the outstanding tutorial of Karpathy [Kar22b].

A.2.1 Differentials

A full accounting of this subsection is given in the excellent guide [BEJ25]. To motivate differentials, let us first consider the simple example of a differentiable function $\mathcal{L}\colon \mathbb{R} \to \mathbb{R}$ acting on a parameter θ. We can write

$$\mathcal{L}(\theta^+) - \mathcal{L}(\theta) = \mathcal{L}'(\theta) \cdot (\theta^+ - \theta) + o(|\theta^+ - \theta|). \tag{A.2.1}$$

If we take $\delta\theta \doteq \theta^+ - \theta$ and $\delta\mathcal{L} \doteq \mathcal{L}(\theta + \delta\theta) - \mathcal{L}(\theta)$, we can write

$$\delta\mathcal{L} = \mathcal{L}'(\theta) \cdot \delta\theta + o(|\delta\theta|). \tag{A.2.2}$$

We will (non-rigorously) define $\mathrm{d}\theta$ and $\mathrm{d}\mathcal{L}$, i.e., the *differentials* of θ and $\mathcal{L}$, to be *infinitesimally small* changes in θ and $\mathcal{L}$. Think of them as what one gets when $\delta\theta$ (and therefore $\delta\mathcal{L}$) are extremely small. The goal of differential calculus, in some sense, is to study the relationships between the differentials $\mathrm{d}\theta$ and $\mathrm{d}\mathcal{L}$, namely, seeing how small changes in the input of a function change the output. We should note that the differential $\mathrm{d}\theta$ is the *same shape* as θ, and the differential $\mathrm{d}\mathcal{L}$ is the *same shape* as $\mathcal{L}$. In particular, we can write

$$\mathrm{d}\mathcal{L} = \mathcal{L}'(\theta) \cdot \mathrm{d}\theta, \tag{A.2.3}$$

whereby we have that all higher powers of $|\mathrm{d}\theta|$, such as $(\mathrm{d}\theta)^2$, are 0.

Let's see how this works for higher dimensions, i.e., $\mathcal{L}\colon \mathbb{R}^n \to \mathbb{R}$. Then we still have

$$\mathrm{d}\mathcal{L} = \mathcal{L}'(\theta) \cdot \mathrm{d}\theta \tag{A.2.4}$$

for some notion of a derivative $\mathcal{L}'(\theta)$. Since θ (hence $\mathrm{d}\theta$) is a column vector here and $\mathcal{L}$ (hence $\mathrm{d}\mathcal{L}$) is a scalar, we must have that $\mathcal{L}'(\theta)$ is a row vector. In this case, $\mathcal{L}'(\theta)$ is the Jacobian of $\mathcal{L}$ w.r.t. θ. Here notice that we have set all higher powers and products of coordinates of $\mathrm{d}\theta$ to 0. In sum,

All products and powers ≥ 2 of differentials are equal to 0.

Now consider a higher-order tensor function $\mathcal{L}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$. Then our basic linearization equation is insufficient for this case: $\mathrm{d}\mathcal{L} = \mathcal{L}'(\theta) \cdot \mathrm{d}\theta$ does not make sense since θ is an $m \times n$ matrix but $\mathrm{d}\mathcal{L}$ is a $p \times q$ matrix, so there is no possible vector or matrix shape for $\mathcal{L}'(\theta)$ that works in general (as no matrix can multiply a $m \times n$ matrix to form a $p \times q$ matrix unless $m = p$). So we must have a slightly more advanced interpretation.

Namely, we consider $\mathcal{L}'(\theta)$ as a *linear transformation* whose input is θ-space and whose output is $\mathcal{L}$-space, which takes in a small change in θ and outputs the corresponding small change in $\mathcal{L}$. Namely, we can write

$$\mathrm{d}\mathcal{L} = \mathcal{L}'(\theta)[\mathrm{d}\theta]. \tag{A.2.5}$$

In the previous cases, $\mathcal{L}'(\theta)$ was first a linear operator $\mathbb{R} \to \mathbb{R}$ whose action was to multiply its input by the scalar derivative of $\mathcal{L}$ with respect to θ, and then a linear operator $\mathbb{R}^n \to \mathbb{R}$ whose action was to multiply its input by the Jacobian derivative of $\mathcal{L}$ with respect to θ. In general $\mathcal{L}'(\theta)$ is the "derivative" of $\mathcal{L}$ w.r.t. θ. Think of $\mathcal{L}'$ as a generalized version of the Jacobian of $\mathcal{L}$. As such, it follows some simple derivative rules, most crucially the chain rule.

Theorem A.1 (Differential Chain Rule)**.** *Suppose $\mathcal{L} = f \circ g$ where f and g are differentiable. Then*

$$\mathrm{d}\mathcal{L} = f'(g(\theta))g'(\theta)[\mathrm{d}\theta], \tag{A.2.6}$$

where (as usual) multiplication indicates composition of linear operators. In particular,

$$\mathcal{L}'(\theta) = f'(g(\theta))g'(\theta) \tag{A.2.7}$$

in the sense of equality of linear operators.

It is productive to think of the multivariate chain rule in functorial terms: composition of functions gets 'turned into' matrix multiplication of Jacobians (composition of linear operators!). We illustrate the power of this result and this perspective through several examples.

Example A.4. Consider the function $f(\boldsymbol{X}) = \boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}\mathbf{1}^\top$. Then

$$\mathrm{d}f = f(\boldsymbol{X} + \mathrm{d}\boldsymbol{X}) - f(\boldsymbol{X}) = [\boldsymbol{W}(\boldsymbol{X} + \mathrm{d}\boldsymbol{X}) + \boldsymbol{b}\mathbf{1}^\top] - [\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}\mathbf{1}^\top] = \boldsymbol{W}\mathrm{d}\boldsymbol{X}. \tag{A.2.8}$$

Thus the derivative of an affine function w.r.t. its input is

$$f'(\boldsymbol{X})[\mathrm{d}\boldsymbol{X}] = \boldsymbol{W}\mathrm{d}\boldsymbol{X} \implies f'(\boldsymbol{X}) = \boldsymbol{W}. \tag{A.2.9}$$

Notice that f' is *constant*. On the other hand, consider the function $g(\boldsymbol{W}, \boldsymbol{b}) = \boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}\mathbf{1}^\top$. Then

$$\begin{aligned} \mathrm{d}g &= g(\boldsymbol{W} + \mathrm{d}\boldsymbol{W}, \boldsymbol{b} + \mathrm{d}\boldsymbol{b}) - g(\boldsymbol{W}, \boldsymbol{b}) && \text{(A.2.10)} \\ &= [(\boldsymbol{W} + \mathrm{d}\boldsymbol{W})\boldsymbol{X} + (\boldsymbol{b} + \mathrm{d}\boldsymbol{b})\mathbf{1}^\top] - [\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}\mathbf{1}^\top] && \text{(A.2.11)} \\ &= (\mathrm{d}\boldsymbol{W})\boldsymbol{X} + (\mathrm{d}\boldsymbol{b})\mathbf{1}^\top = g'(\boldsymbol{W}, \boldsymbol{b})[\mathrm{d}\boldsymbol{W}, \mathrm{d}\boldsymbol{b}]. && \text{(A.2.12)} \end{aligned}$$

Notice that this derivative is *constant* in $\boldsymbol{W}, \boldsymbol{b}$ (which makes sense since g itself is linear) and linear in the differential inputs $\mathrm{d}\boldsymbol{W}, \mathrm{d}\boldsymbol{b}$. ■

Example A.5. Consider the function $f = gh$ where g, h are differentiable functions whose outputs can multiply together. Then $f = p \circ v$ where $v = (g, h)$ and $p(a, b) = ab$. Applying the chain rule we have

$$\mathrm{d}f = p'(v(x))v'(x)[\mathrm{d}x]. \tag{A.2.13}$$

To compute $v'(x)$ we can compute

$$\mathrm{d}v = v'(x)[\mathrm{d}x] = v(x + \mathrm{d}x) - v(x) = \begin{bmatrix} g(x + \mathrm{d}x) - g(x) \\ h(x + \mathrm{d}x) - h(x) \end{bmatrix} = \begin{bmatrix} g'(x)[\mathrm{d}x] \\ h'(x)[\mathrm{d}x] \end{bmatrix}. \tag{A.2.14}$$

To compute p' we can compute

$$\mathrm{d}p = p'(a,b)[\mathrm{d}a,\mathrm{d}b] = p(a+\mathrm{d}a, b+\mathrm{d}b) - p(a,b) = (a+\mathrm{d}a)(b+\mathrm{d}b) - ab \tag{A.2.15}$$

$$= (\mathrm{d}a)b + a(\mathrm{d}b) + (\mathrm{d}a)(\mathrm{d}b) = (\mathrm{d}a)b + a(\mathrm{d}b), \tag{A.2.16}$$

where (recall) the product of the differentials $\mathrm{d}a$ and $\mathrm{d}b$ is set to 0. Therefore

$$p'(a,b)[\mathrm{d}a,\mathrm{d}b] = (\mathrm{d}a)b + (\mathrm{d}b)a. \tag{A.2.17}$$

Putting these together, we find

$$f'(x)[\mathrm{d}x] = p'(v(x))v'(x)[\mathrm{d}x] = p'(g(x),h(x))[g'(x)[\mathrm{d}x], h'(x)[\mathrm{d}x]] \tag{A.2.18}$$

$$= (g'(x)[\mathrm{d}x])h(x) + g(x)(h'(x)[\mathrm{d}x]). \tag{A.2.19}$$

This gives

$$f'(x)[\mathrm{d}x] = (g'(x)[\mathrm{d}x])h(x) + g(x)(h'(x)[\mathrm{d}x]). \tag{A.2.20}$$

If for example we say that $f,g,h\colon \mathbb{R}\to\mathbb{R}$ then everything commutes so

$$f'(x)[\mathrm{d}x] = (g'(x)h(x) + g(x)h'(x))[\mathrm{d}x] \implies f'(x) = g'(x)h(x) + g(x)h'(x) \tag{A.2.21}$$

which is the familiar product rule. ■

Example A.6. Consider the function $f(\boldsymbol{A}) = \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}$ where $\boldsymbol{A}$ is a matrix and $\boldsymbol{B}$ is a constant matrix. Then, letting $f = gh$ where $g(\boldsymbol{A}) = \boldsymbol{A}^\top\boldsymbol{A}$ and $h(\boldsymbol{A}) = \boldsymbol{B}\boldsymbol{A}$, we can use the product rule to obtain

$$f'(\boldsymbol{A})[\mathrm{d}\boldsymbol{A}] = (g'(\boldsymbol{A})[\mathrm{d}\boldsymbol{A}])h(\boldsymbol{A}) + g(\boldsymbol{A})(h'(\boldsymbol{A})[\mathrm{d}\boldsymbol{A}]) \tag{A.2.22}$$

$$= ((\mathrm{d}\boldsymbol{A})^\top\boldsymbol{A} + \boldsymbol{A}^\top(\mathrm{d}\boldsymbol{A}))\boldsymbol{B}\boldsymbol{A} + \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{B}(\mathrm{d}\boldsymbol{A}). \tag{A.2.23}$$

■

Example A.7. Consider the function $f\colon \mathbb{R}^{m\times n\times k}\to\mathbb{R}^{m\times n}$ given by

$$f(\boldsymbol{A})_{ij} = \sum_{t=1}^{k} A_{ijt}. \tag{A.2.24}$$

We cannot write a (matrix-valued) Jacobian or gradient for this function. But we can compute its differential just fine:

$$\mathrm{d}f_{ij} = [f(\boldsymbol{A}+\mathrm{d}\boldsymbol{A}) - f(\boldsymbol{A})]_{ij} = \sum_{t=1}^{k} \mathrm{d}\boldsymbol{A}_{ijt} = \mathbf{1}_k^\top(\mathrm{d}\boldsymbol{A})_{ij}. \tag{A.2.25}$$

So

$$(f'(\boldsymbol{A})[\mathrm{d}\boldsymbol{A}])_{ij} = \mathbf{1}_k^\top(\mathrm{d}\boldsymbol{A})_{ij}, \tag{A.2.26}$$

which represents a higher-order tensor multiplication operation that is nonetheless well-defined. ■

This gives us all the technology we need to compute differentials of everything. The last thing we cover in this section is a method to compute gradients using the differential. Namely, for a function $\mathcal{L}$ whose output is a scalar, the gradient $\nabla\mathcal{L}$ is defined as

$$\mathrm{d}\mathcal{L} = \mathcal{L}'(\theta)[\mathrm{d}\theta] = \langle \nabla\mathcal{L}(\theta), \mathrm{d}\theta \rangle, \tag{A.2.27}$$

where the inner product here is the "standard" inner product for the specified objects (i.e., for vectors it's the ℓ^2 inner product, whereas for matrices it's the Frobenius inner product, and for higher-order tensors it's the analogous sum-of-coordinates inner product). This definition is the correct generalization of the 'familiar' example of the gradient of a function from $\mathbb{R}^n$ to $\mathbb{R}$ as the vector of partial derivatives—a version of Taylor's theorem for general functions $f : \mathbb{R}^m \to \mathbb{R}^n$ makes this connection rigorous. So one way to compute the gradient $\nabla\mathcal{L}$ is to compute the differential $\mathrm{d}\mathcal{L}$ and rewrite it in the form $\langle \text{something}, \mathrm{d}\theta \rangle$, then that "something" is the gradient.

A.2.2 Automatic Differentiation

The main idea of AD is to compute the chain rule efficiently. The basic problem we need to cope with is the following. In the optimization section of the appendix, we considered that the parameter space Θ was an abstract Euclidean space like $\mathbb{R}^n$. In practice the parameters are really some collection of vectors, matrices, and higher-order objects: $\Theta = \mathbb{R}^{m\times n} \times \mathbb{R}^n \times \mathbb{R}^{r\times q\times p} \times \mathbb{R}^{r\times q} \times \cdots$. While in theory this is the same thing as a large parameter space $\mathbb{R}^{n'}$ for some (very large) n', computationally efficient algorithms for differentiation must treat these two spaces differently. Forward and reverse mode automatic differentiation are two different schemes for performing this computation.

Let us do a simple example to start. Let $\mathcal{L}$ be defined by $\mathcal{L} = a \circ b \circ c$ where a, b, c are differentiable. Then the *chain rule* gives

$$\mathcal{L}'(\theta) = a'(b(c(\theta)))b'(c(\theta))c'(\theta). \tag{A.2.28}$$

To compute $\mathcal{L}(\theta)$, we first compute $c(\theta)$ then $b(c(\theta))$ then $a(b(c(\theta)))$, and store them all. There are two ways to compute $\mathcal{L}'(\theta)$. The *forward-mode AD* will compute

$$c'(\theta) \implies b'(c(\theta))c'(\theta) \implies a'(b(c(\theta)))b'(c(\theta))c'(\theta) \tag{A.2.29}$$

i.e., computing the derivatives "from the bottom-up". The *reverse mode AD* will compute

$$a'(b(c(\theta))) \implies a'(b(c(\theta)))b'(c(\theta)) \implies a'(b(c(\theta)))b'(c(\theta))c'(\theta), \tag{A.2.30}$$

i.e., computing the derivatives "from the top down". To see why this matters, suppose that $f\colon \mathbb{R}^p \to \mathbb{R}^s$ is given by $f = a \circ b \circ c$ where $a\colon \mathbb{R}^r \to \mathbb{R}^s$, $b\colon \mathbb{R}^q \to \mathbb{R}^r$, $c\colon \mathbb{R}^p \to \mathbb{R}^q$. Then the chain rule is:

$$f'(\boldsymbol{x}) = a'(b(c(\boldsymbol{x})))b'(c(\boldsymbol{x}))c'(\boldsymbol{x}) \tag{A.2.31}$$

where (recall) f' is the derivative, in this case the Jacobian (since the input and output of each function are both vectors). Assuming that computing each Jacobian is trivial and the only cost is multiplying the Jacobians together, forward-mode AD has the following computational costs (assuming that multiplying $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times k}$ takes $\mathcal{O}(mnk)$ time):

$$\text{computing } c'(\boldsymbol{x}) \in \mathbb{R}^{q\times p} \text{ takes negligible time} \tag{A.2.32}$$

$$\text{computing } b'(c(\boldsymbol{x}))c'(x) \in \mathbb{R}^{r\times p} \text{ takes } \mathcal{O}(pqr) \text{ time} \tag{A.2.33}$$

$$\text{computing } a'(b(c(\boldsymbol{x})))b'(c(\boldsymbol{x}))c'(\boldsymbol{x}) \in \mathbb{R}^{s\times p} \text{ takes } \mathcal{O}(pqr + prs) \text{ time.} \tag{A.2.34}$$

Meanwhile, doing reverse-mode AD has the following computational costs:

$$\text{computing } a'(b(c(\boldsymbol{x}))) \in \mathbb{R}^{s\times r} \text{ takes negligible time} \tag{A.2.35}$$

$$\text{computing } a'(b(c(\boldsymbol{x})))b'(c(\boldsymbol{x})) \in \mathbb{R}^{s\times q} \text{ takes } \mathcal{O}(qrs) \text{ time} \tag{A.2.36}$$

$$\text{computing } a'(b(c(\boldsymbol{x})))b'(c(\boldsymbol{x}))c'(\boldsymbol{x}) \in \mathbb{R}^{s\times p} \text{ takes } \mathcal{O}(qrs + pqs) \text{ time.} \tag{A.2.37}$$

In other words, the forward-mode AD takes $\mathcal{O}(p(qr+rs))$ time, and the reverse-mode AD takes $\mathcal{O}(s(pq + qr))$ time. *These take a different amount of time!*

More generally, suppose that $f = f^L \circ \cdots \circ f^1$ where each $f^\ell \colon \mathbb{R}^{d^{\ell-1}} \to \mathbb{R}^{d^\ell}$, so that $f \colon \mathbb{R}^{d^0} \to \mathbb{R}^{d^L}$. Then the forward-mode AD takes $\mathcal{O}(d^0(\sum_{\ell=2}^{L} d^{\ell-1}d^\ell))$ time while the reverse-mode AD takes $\mathcal{O}(d^L(\sum_{\ell=1}^{L-1} d^{\ell-1}d^\ell))$ time. From the above rates, we see that all else equal:

- If the function to optimize has *more outputs than inputs* (i.e., $d^L > d^0$), use *forward-mode AD*.
- If the function to optimize has *more inputs than outputs* (i.e., $d^0 > d^L$), use *reverse-mode AD*.

In a neural network, we compute the gradient of a *loss function* $\mathcal{L} \colon \Theta \to \mathbb{R}$, where the parameter space Θ is usually very high-dimensional. So in practice we always use reverse-mode AD for training neural networks. Reverse-mode AD, in the context of training neural networks, is called *backpropagation*.

A.2.3 Back Propagation

In this section, we will discuss algorithmic backpropagation using a simple yet completely practical example. Suppose that we fix an input-label pair $(\boldsymbol{X}, \boldsymbol{y})$, and fix a network architecture $f_\theta = f_\theta^L \circ \cdots \circ f_\theta^1 \circ f_\theta^{\text{emb}}$ where $\theta = (\theta^{\text{emb}}, \theta^1, \ldots, \theta^L, \theta^{\text{head}})$ and task-specific head $h_{\theta^{\text{head}}}$, and write

$$\boldsymbol{Z}_\theta^1(\boldsymbol{X}) \doteq f_\theta^{\text{emb}}(\boldsymbol{X}), \tag{A.2.38}$$

$$\boldsymbol{Z}_\theta^{\ell+1}(\boldsymbol{X}) \doteq f_\theta^\ell(\boldsymbol{Z}_\theta^\ell(\boldsymbol{X})), \quad \forall \ell \in \{1, \ldots, L\}, \tag{A.2.39}$$

$$\hat{\boldsymbol{y}}_{\theta^{\text{head}}}(\boldsymbol{X}) \doteq h_{\theta^{\text{head}}}(\boldsymbol{Z}_\theta^{L+1}(\boldsymbol{X})). \tag{A.2.40}$$

Then, we can define the loss on this one term by

$$\mathcal{L}(\theta) \doteq \mathsf{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}_\theta(\boldsymbol{X})), \tag{A.2.41}$$

where L is a differentiable function of its second argument. Then the backward-mode AD instructs us to compute the derivatives in the order $\theta^{\text{head}}, \theta^L, \dots, \theta^1, \theta^{\text{emb}}$.

To carry out this computation, let us make some changes to the notation.

- First, let us change the notation to emphasize the dependency structure between the variables. Namely,

$$\boldsymbol{Z}^1 \doteq f^{\text{emb}}(\boldsymbol{X}, \theta^{\text{emb}}) \tag{A.2.42}$$

$$\boldsymbol{Z}^{\ell+1} \doteq f^\ell(\boldsymbol{Z}^\ell, \theta^\ell), \qquad \forall \ell \in \{1, \dots, L\}, \tag{A.2.43}$$

$$\hat{\boldsymbol{y}} \doteq h(\boldsymbol{Z}^{L+1}, \theta^{\text{head}}), \tag{A.2.44}$$

$$\mathcal{L} \doteq \mathsf{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}). \tag{A.2.45}$$

- Then, instead of having the derivative be f', we explicitly notate the independent variable and write the derivative as $\frac{\mathrm{d}f}{\mathrm{d}\theta^1}$, for example. This is because there are many variables in our model and we only care about one at a time.

We can start by computing the appropriate differentials. First, for θ^{head} we have

$$\mathrm{d}\mathcal{L} = \mathrm{d}\mathsf{L} \tag{A.2.46}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \mathrm{d}\hat{\boldsymbol{y}} \tag{A.2.47}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \mathrm{d}(h(\boldsymbol{Z}^{L+1}, \theta^{\text{head}})) \tag{A.2.48}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \left[\frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{Z}^{L+1}} \cdot \mathrm{d}\boldsymbol{Z}^{L+1} + \frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \cdot \mathrm{d}\theta^{\text{head}} \right]. \tag{A.2.49}$$

Now since $\boldsymbol{Z}^{L+1}$ does not depend on θ^{head}, we have $\mathrm{d}\boldsymbol{Z}^{L+1} = 0$, so in the end it holds (using the fact that the gradient is the transpose of the derivative for a function L whose codomain is $\mathbb{R}$):

$$\mathrm{d}\mathcal{L} = \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \cdot \mathrm{d}\theta^{\text{head}} \tag{A.2.50}$$

$$= [\nabla_{\hat{\boldsymbol{y}}} \mathsf{L}]^\top \frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \cdot \mathrm{d}\theta^{\text{head}} \tag{A.2.51}$$

$$= \left\langle \nabla_{\hat{\boldsymbol{y}}} \mathsf{L}, \frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \cdot \mathrm{d}\theta^{\text{head}} \right\rangle \tag{A.2.52}$$

$$= \left\langle \left(\frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \right)^* \nabla_{\hat{\boldsymbol{y}}} \mathsf{L}, \mathrm{d}\theta^{\text{head}} \right\rangle \tag{A.2.53}$$

$$= \langle \nabla_{\theta^{\text{head}}} \mathcal{L}, \mathrm{d}\theta^{\text{head}} \rangle. \tag{A.2.54}$$

Thus to compute $\nabla_{\theta^{\text{head}}}\mathcal{L}$, we compute the gradient $\nabla_{\hat{\boldsymbol{y}}}\mathsf{L}$ and the *adjoint*[6] of the derivative $\frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}}$ and multiply (i.e., apply the adjoint linear transformation to the gradient). In practice, both derivatives can be computed by hand, but many modern computational frameworks can *automatically* define the derivatives (and/or their adjoints) given code for the "forward pass," i.e., the loss function computation. While extending this automatic derivative definition to as many functions as possible is an area of active research, the resource [BEJ25] describes one basic approach to do it in some detail. By the way, backpropagation is also called the *adjoint method* for this reason — i.e., that we use adjoint derivatives to compute the gradient.

Now let us compute the differentials w.r.t. some θ^ℓ:

$$\begin{aligned}
\mathrm{d}\mathcal{L} &= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \mathrm{d}\boldsymbol{Z}^{\ell+1} && \text{(A.2.55)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \mathrm{d}(f^\ell(\boldsymbol{Z}^\ell, \theta^\ell)) && \text{(A.2.56)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \left[\frac{\mathrm{d}f^\ell}{\mathrm{d}\boldsymbol{Z}^\ell} \cdot \mathrm{d}\boldsymbol{Z}^\ell + \frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell} \cdot \mathrm{d}\theta^\ell\right] && \text{(A.2.57)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell} \cdot \mathrm{d}\theta^\ell \qquad (\text{b/c } \boldsymbol{Z}^\ell \text{ isn't fn. of } \theta^\ell \text{ so } \mathrm{d}\boldsymbol{Z}^\ell = 0) && \text{(A.2.58)}\\
&= [\nabla_{\boldsymbol{Z}^{\ell+1}}\mathcal{L}]^\top \frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell} \cdot \mathrm{d}\theta^\ell && \text{(A.2.59)}\\
&= \left\langle \nabla_{\boldsymbol{Z}^{\ell+1}}\mathcal{L}, \frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell} \cdot \mathrm{d}\theta^\ell \right\rangle && \text{(A.2.60)}\\
&= \left\langle \left(\frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell}\right)^* \nabla_{\boldsymbol{Z}^{\ell+1}}\mathcal{L}, \mathrm{d}\theta^\ell \right\rangle && \text{(A.2.61)}\\
&= \langle \nabla_{\theta^\ell}\mathcal{L}, \mathrm{d}\theta^\ell \rangle. && \text{(A.2.62)}
\end{aligned}$$

Thus to compute $\nabla_{\theta^\ell}\mathcal{L}$ we compute $\nabla_{\boldsymbol{Z}^{\ell+1}}\mathcal{L}$ then apply the adjoint derivative $\left(\frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell}\right)^*$ to it. Since $\nabla_{\theta^\ell}\mathcal{L}$ depends on $\nabla_{\boldsymbol{Z}^{\ell+1}}\mathcal{L}$, we also want to be able to compute the gradients w.r.t. $\boldsymbol{Z}^\ell$. This can be computed in the exact same way:

$$\begin{aligned}
\mathrm{d}\mathcal{L} &= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \mathrm{d}\boldsymbol{Z}^{\ell+1} && \text{(A.2.63)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \mathrm{d}(f^\ell(\boldsymbol{Z}^\ell, \theta^\ell)) && \text{(A.2.64)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \left[\frac{\mathrm{d}f^\ell}{\mathrm{d}\boldsymbol{Z}^\ell} \cdot \mathrm{d}\boldsymbol{Z}^\ell + \frac{\mathrm{d}f^\ell}{\mathrm{d}\theta^\ell} \cdot \mathrm{d}\theta^\ell\right] && \text{(A.2.65)}\\
&= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{Z}^{\ell+1}} \cdot \frac{\mathrm{d}f^\ell}{\mathrm{d}\boldsymbol{Z}^\ell} \cdot \mathrm{d}\boldsymbol{Z}^\ell \qquad (\text{b/c } \theta^\ell \text{ isn't fn. of } \boldsymbol{Z}^\ell \text{ so } \mathrm{d}\theta^\ell = 0 \text{ this time}) && \text{(A.2.66)}
\end{aligned}$$

[6] The adjoint is like a generalized transpose for more general linear transformations. Particularly, for a given pair of inner product spaces and linear transformation T between those spaces, the adjoint T^* is defined by the identity $\langle T\boldsymbol{x}, \boldsymbol{y}\rangle = \langle \boldsymbol{x}, T^*\boldsymbol{y}\rangle$. In finite dimensions (i.e., all cases relevant to this book) the adjoint exists and is unique.

$$= \left\langle \left(\frac{\mathrm{d}f^\ell}{\mathrm{d}\boldsymbol{Z}^\ell} \right)^* \nabla_{\boldsymbol{Z}^{\ell+1}} \mathcal{L}, \mathrm{d}\boldsymbol{Z}^\ell \right\rangle \qquad \text{(same machine)} \tag{A.2.67}$$

$$= \langle \nabla_{\boldsymbol{Z}^\ell} \mathcal{L}, \mathrm{d}\boldsymbol{Z}^\ell \rangle. \tag{A.2.68}$$

Thus to compute $\nabla_{\boldsymbol{Z}^\ell} \mathcal{L}$ we compute $\nabla_{\boldsymbol{Z}^{\ell+1}} \mathcal{L}$ then apply the adjoint derivative $\left(\frac{\mathrm{d}f^\ell}{\mathrm{d}\boldsymbol{Z}^\ell} \right)^*$ to it. So we have a recursion to compute $\nabla_{\boldsymbol{Z}^\ell} \mathcal{L}$ for all ℓ, with base case $\nabla_{\boldsymbol{Z}^{L+1}} \mathcal{L}$, which is given by

$$\mathrm{d}\mathcal{L} = \mathrm{d}\mathsf{L} \tag{A.2.69}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \mathrm{d}\hat{\boldsymbol{y}} \tag{A.2.70}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \mathrm{d}h(\boldsymbol{Z}^{L+1}, \theta^{\text{head}}) \tag{A.2.71}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \left[\frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{Z}^{L+1}} \cdot \mathrm{d}\boldsymbol{Z}^{L+1} + \frac{\mathrm{d}h}{\mathrm{d}\theta^{\text{head}}} \cdot \mathrm{d}\theta^{\text{head}} \right] \tag{A.2.72}$$

$$= \frac{\mathrm{d}\mathsf{L}}{\mathrm{d}\hat{\boldsymbol{y}}} \cdot \frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{Z}^{L+1}} \cdot \mathrm{d}\boldsymbol{Z}^{L+1} \tag{A.2.73}$$

$$= \left\langle \left(\frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{Z}^{L+1}} \right)^* \nabla_{\hat{\boldsymbol{y}}} \mathsf{L}, \mathrm{d}\boldsymbol{Z}^{L+1} \right\rangle \tag{A.2.74}$$

$$= \langle \nabla_{\boldsymbol{Z}^{L+1}} \mathcal{L}, \mathrm{d}\boldsymbol{Z}^{L+1} \rangle. \tag{A.2.75}$$

Thus we have the recursion:

$$\nabla_{\boldsymbol{Z}^{L+1}} \mathcal{L} = \left(\frac{\mathrm{d}h}{\mathrm{d}\boldsymbol{Z}^{L+1}} \right)^* \nabla_{\hat{\boldsymbol{y}}} \mathsf{L} \tag{A.2.76}$$

$$\nabla_{\boldsymbol{Z}^L} \mathcal{L} = \left(\frac{\mathrm{d}f^L}{\mathrm{d}\boldsymbol{Z}^L} \right)^* \nabla_{\boldsymbol{Z}^{L+1}} \mathcal{L} \tag{A.2.77}$$

$$\nabla_{\theta^L} \mathcal{L} = \left(\frac{\mathrm{d}f^L}{\mathrm{d}\theta^L} \right)^* \nabla_{\boldsymbol{Z}^{L+1}} \mathcal{L} \tag{A.2.78}$$

$$\vdots \tag{A.2.79}$$

$$\nabla_{\boldsymbol{Z}^1} \mathcal{L} = \left(\frac{\mathrm{d}f^1}{\mathrm{d}\boldsymbol{Z}^1} \right)^* \nabla_{\boldsymbol{Z}^2} \mathcal{L} \tag{A.2.80}$$

$$\nabla_{\theta^1} \mathcal{L} = \left(\frac{\mathrm{d}f^1}{\mathrm{d}\theta^1} \right)^* \nabla_{\boldsymbol{Z}^2} \mathcal{L} \tag{A.2.81}$$

$$\nabla_{\theta^{\text{emb}}} \mathcal{L} = \left(\frac{\mathrm{d}f^{\text{emb}}}{\mathrm{d}\theta^{\text{emb}}} \right)^* \nabla_{\boldsymbol{Z}^1} \mathcal{L}. \tag{A.2.82}$$

This gives us a computationally efficient algorithm to find all gradients in the whole network.

We'll finish this section by computing the adjoint derivative for a simple layer.

Example A.8. Consider the "linear" (affine) layer f^ℓ

$$f^\ell(\boldsymbol{Z}, \boldsymbol{W}^\ell, \boldsymbol{b}^\ell) \doteq \boldsymbol{W}^\ell \boldsymbol{Z} + \boldsymbol{b}^\ell \boldsymbol{1}^\top = \begin{bmatrix} \boldsymbol{W}^\ell & \boldsymbol{b}^\ell \end{bmatrix}. \tag{A.2.83}$$

We can compute the differential w.r.t. both parameters as

$$\begin{aligned} \mathrm{d}f^\ell &= [(\boldsymbol{W}^\ell + \mathrm{d}\boldsymbol{W}^\ell)\boldsymbol{Z} + (\boldsymbol{b}^\ell + \mathrm{d}\boldsymbol{b}^\ell)\boldsymbol{1}^\top] - [\boldsymbol{W}^\ell \boldsymbol{Z} + \boldsymbol{b}^\ell \boldsymbol{1}^\top] && \text{(A.2.84)} \\ &= (\mathrm{d}\boldsymbol{W}^\ell)\boldsymbol{Z} + (\mathrm{d}\boldsymbol{b}^\ell)\boldsymbol{1}^\top. && \text{(A.2.85)} \end{aligned}$$

Thus the derivative of this transformation is

$$\frac{\mathrm{d}f^\ell}{\mathrm{d}(\boldsymbol{W}^\ell, \boldsymbol{b}^\ell)}[\mathrm{d}\boldsymbol{W}^\ell, \mathrm{d}\boldsymbol{b}^\ell] = (\mathrm{d}\boldsymbol{W}^\ell)\boldsymbol{Z} + (\mathrm{d}\boldsymbol{b}^\ell)\boldsymbol{1}^\top, \tag{A.2.86}$$

namely, representing the following linear transformation from $\mathbb{R}^{m\times d} \times \mathbb{R}^m$ to $\mathbb{R}^{m\times n}$:

$$T[\boldsymbol{A}, \boldsymbol{u}] = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{u}\boldsymbol{1}^\top. \tag{A.2.87}$$

We calculate the adjoint $T^*\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times d}\times\mathbb{R}^m$ w.r.t. the sum-over-coordinates (Frobenius) inner product by the following procedure:

$$\begin{aligned} \langle T[\boldsymbol{A}, \boldsymbol{u}], \boldsymbol{B}\rangle_{\mathbb{R}^{m\times n}} &= \operatorname{tr}((\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{u}\boldsymbol{1}^\top)\boldsymbol{B}^\top) && \text{(A.2.88)} \\ &= \operatorname{tr}(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top + \boldsymbol{u}\boldsymbol{1}^\top\boldsymbol{B}^\top) && \text{(A.2.89)} \\ &= \operatorname{tr}(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top) + \operatorname{tr}(\boldsymbol{u}\boldsymbol{1}^\top\boldsymbol{B}^\top) && \text{(A.2.90)} \\ &= \operatorname{tr}(\boldsymbol{B}\boldsymbol{Z}^\top\boldsymbol{A}^\top) + \operatorname{tr}(\boldsymbol{1}^\top\boldsymbol{B}^\top\boldsymbol{u}) && \text{(A.2.91)} \\ &= \langle \boldsymbol{B}\boldsymbol{Z}^\top, \boldsymbol{A}\rangle_{\mathbb{R}^{m\times d}} + \langle \boldsymbol{B}\boldsymbol{1}, \boldsymbol{u}\rangle_{\mathbb{R}^m} && \text{(A.2.92)} \\ &= \langle \boldsymbol{B}(\boldsymbol{Z}^\top, \boldsymbol{1}), (\boldsymbol{A}, \boldsymbol{u})\rangle_{\mathbb{R}^{m\times d}\times\mathbb{R}^m} && \text{(A.2.93)} \end{aligned}$$

So $T^*\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{Z}^\top, \boldsymbol{1})$. ■

Note that as a simple application of chain rule, both backpropagation and automatic differentiation work over general "computational graphs", i.e., compositions of (simple) functions. We give all examples as neural network layers because this is the most common example in practice.

A.3 Game Theory and Minimax Optimization

In certain cases, such as in Chapter 6, a learning problem cannot be reduced to a single optimization problem but rather represents multiple potentially opposing components of the system trying to each minimize their own objective. Examples of this paradigm include distribution learning via generative adversarial networks (GAN) and closed-loop transcription (CTRL). We will denote such a system as a *two-player game*, where we have two "players" (i.e., components) called Player 1 and Player 2 trying to minimize their objectives $\mathcal{L}^1$ and $\mathcal{L}^2$ respectively. Player 1 picks parameters $\theta \in \Theta$ and Player 2 picks parameters

$\eta \in \mathrm{H}$. In this book we consider the special case of *zero-sum games*, i.e., defining a common objective $\mathcal{L}$ such that $\mathcal{L} = -\mathcal{L}^1 = \mathcal{L}^2$.

Our first, very preliminary example is as follows. Suppose that there exists functions $u(\theta)$ and $v(\eta)$ such that

$$\mathcal{L}(\theta, \eta) = -u(\theta) + v(\eta). \tag{A.3.1}$$

Then both players' objectives are independent of the other player, and the players should try to achieve their respective optima:

$$\theta^\star \in \arg\min_{\theta\in\Theta} u(\theta), \qquad \eta^\star \in \arg\min_{\eta\in\mathrm{H}} v(\eta). \tag{A.3.2}$$

The pair $(\theta^\star, \eta^\star)$ is a straightforward special case of an *equilibrium*: a situation where neither player will want to move, given the chance, since moving will end up making their own situation worse. However, not all games are so trivial; many have more complicated objectives and information structures.

In this book, the relevant game-theoretic formalism is a Stackelberg game (variously called sequential game). In this formalism, one player (without loss of generality Player 1, and also described as a *leader*) picks their parameters before the other (i.e., Player 2, also described as a *follower*), and the follower can use the full knowledge of the leader's choice to make their own choice. The correct notion of equilibrium for a Stackelberg game is a *Stackelberg equilibrium*. To explain this equilibrium, note that since Player 2 (i.e., the follower) can choose η reactively to the choice θ_1 made by Player 1 (i.e., the leader), Player 2 would of course choose the η which minimizes $\mathcal{L}(\theta_1, \cdot)$. But of course a rational Player 1 would realize this, and so pick a θ_1 such that the worst-case η picked by Player 2 according to this rule is not too bad. More formally, let $\mathcal{S}(\theta) \doteq \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta)$ be the set of η minimizing $\mathcal{L}(\theta, \eta)$, i.e., the set of all η which Player 2 is liable to play given that Player 1 has played θ. Then $(\theta^\star, \eta^\star)$ is a Stackelberg equilibrium if

$$\theta^\star \in \arg\max_{\theta\in\Theta} \min_{\eta\in\mathcal{S}(\theta)} \mathcal{L}(\theta, \eta), \qquad \eta^\star \in \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta^\star, \eta). \tag{A.3.3}$$

Actually (proof as exercise), one can show that in the context of two-player zero-sum Stackelberg games, $(\theta^\star, \eta^\star)$ is a Stackelberg equilibrium if and only if

$$\theta^\star \in \arg\max_{\theta\in\Theta} \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta), \qquad \eta^\star \in \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta^\star, \eta), \tag{A.3.4}$$

(note that the notation $\mathcal{S}(\theta)$ is not used nor needed).

Proof. Note that

$$\mathcal{S}(\theta) = \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta), \tag{A.3.5}$$

so it holds

$$\min_{\eta\in\mathcal{S}(\theta)} \mathcal{L}(\theta, \eta) = \min_{\eta\in\arg\,\min_{\eta'\in\mathrm{H}} \mathcal{L}(\theta,\eta')} \mathcal{L}(\theta, \eta) = \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta). \tag{A.3.6}$$

□

In the rest of the section, we will briefly discuss some algorithmic approaches to learn Stackelberg equilibria. The intuition you should have is that learning an equilibrium is like letting the different parts of the system automatically figure out tradeoffs between the different objectives they want to optimize.

We end this section with a caveat: in two-player zero-sum games, if it holds that

$$\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta)=\min_{\eta\in\mathrm{H}}\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta) \tag{A.3.7}$$

then every Stackelberg equilibrium is a saddle point,[7] i.e.,

$$\theta^\star\in\arg\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta^\star),\qquad \eta^\star\in\arg\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta^\star,\eta), \tag{A.3.8}$$

and vice versa, and furthermore each Stackelberg equilibrium has the (same) objective value

$$\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta).$$

Proof. Suppose that indeed

$$\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta)=\min_{\eta\in\mathrm{H}}\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta). \tag{A.3.9}$$

First suppose that $(\theta^\star,\eta^\star)$ is a saddle point. We will show it is a Stackelberg equilibrium. By definition we have

$$\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta^\star,\eta)=\mathcal{L}(\theta^\star,\eta^\star)=\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta^\star). \tag{A.3.10}$$

It then holds for any θ and η that

$$\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta)\le\mathcal{L}(\theta,\eta^\star)\le\mathcal{L}(\theta^\star,\eta^\star). \tag{A.3.11}$$

Therefore

$$\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta)\le\mathcal{L}(\theta^\star,\eta^\star). \tag{A.3.12}$$

Completely symmetrically,

$$\mathcal{L}(\theta^\star,\eta^\star)\le\mathcal{L}(\theta^\star,\eta)\le\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta)\implies\mathcal{L}(\theta^\star,\eta^\star)\le\min_{\eta\in\mathrm{H}}\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta). \tag{A.3.13}$$

Therefore since $\max\min=\min\max$ we have

$$\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta)=\mathcal{L}(\theta^\star,\eta^\star)=\min_{\eta\in\mathrm{H}}\max_{\theta\in\Theta}\mathcal{L}(\theta,\eta). \tag{A.3.14}$$

In particular, it holds that

$$\theta^\star\in\arg\max_{\theta\in\Theta}\min_{\eta\in\mathrm{H}}\mathcal{L}(\theta,\eta). \tag{A.3.15}$$

[7] Famously called a *Nash equilibrium*.

From the saddle point condition we have $\eta^\star \in \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta^\star, \eta)$. So $(\theta^\star, \eta^\star)$ is a Stackelberg equilibrium. Furthermore we have also proved that all saddle points obey (A.3.14).

Now let $(\theta^\star, \eta^\star)$ be a Stackelberg equilibrium. We claim that it is a saddle point, which completes the proof. By the definition of minimax equilibrium,

$$\max_{\theta\in\Theta} \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta) = \mathcal{L}(\theta^\star, \eta^\star). \tag{A.3.16}$$

Then by the $\min\max = \max\min$ assumption we have

$$\max_{\theta\in\Theta} \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta) = \mathcal{L}(\theta^\star, \eta^\star) = \min_{\eta\in\mathrm{H}} \max_{\theta\in\Theta} \mathcal{L}(\theta, \eta). \tag{A.3.17}$$

This proves that all minimax equilibria have the desired objective value $\mathcal{L}(\theta^\star, \eta^\star)$. Now we want to show that

$$\theta^\star \in \arg\max_{\theta\in\Theta} \mathcal{L}(\theta, \eta^\star), \qquad \eta^\star \in \arg\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta^\star, \eta). \tag{A.3.18}$$

Indeed the latter assertion holds by definition of the minimax equilibrium, so only the former need be proved. Namely, we will show that

$$\max_{\theta\in\Theta} \mathcal{L}(\theta, \eta^\star) = \mathcal{L}(\theta^\star, \eta^\star). \tag{A.3.19}$$

To show this note that by definition of the max

$$\mathcal{L}(\theta^\star, \eta^\star) \leq \max_{\theta\in\Theta} \mathcal{L}(\theta, \eta^\star), \tag{A.3.20}$$

meanwhile we have $\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta) \leq \mathcal{L}(\theta, \eta^\star)$ so

$$\mathcal{L}(\theta^\star, \eta^\star) = \max_{\theta\in\Theta} \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta) \leq \max_{\theta\in\Theta} \mathcal{L}(\theta, \eta^\star). \tag{A.3.21}$$

Therefore it holds

$$\mathcal{L}(\theta^\star, \eta^\star) = \max_{\theta\in\Theta} \mathcal{L}(\theta, \eta^\star), \tag{A.3.22}$$

and the proof is complete. □

Conditions under which the $\min\max = \max\min$ equality holds are given by so-called *minimax theorems*; the most famous of these is a theorem of von Neumann. However, in the cases we think about, this property usually does not hold.

A.3.1 Learning Stackelberg Equilibria

How can we learn Stackelberg equilibria via GDA? In general this is clearly impossible, since learning Stackelberg equilibria via GDA is obviously at least as hard as computing a global minimizer of a loss function (say by setting the shared objective $\mathcal{L}(\theta, \eta)$ to only be a function of η). As such, we can achieve two types of convergence guarantees:

- When $\mathcal{L}$ is (strongly) concave in the first argument θ and (strongly) convex in the second argument η (as well as having Lipschitz gradients in both arguments), we can achieve exponentially fast convergence to a Stackelberg equilibrium.
- When $\mathcal{L}$ is not concave or convex in either argument, we can achieve *local convergence guarantees*: namely, if we initialize the parameter values near a (local) Stackelberg equilibrium and the optimization geometry is good then we can learn that equilibrium efficiently.

The former situation is exactly analogous to the case of single-player optimization, where we proved that gradient descent converges exponentially fast for strongly convex objectives which have Lipschitz gradient. The latter situation is also analogous to the case of single-player optimization, although we did not cover it in depth due to technical difficulty; indeed there exist local convergence guarantees for nonconvex objectives which have locally nice geometry.

The algorithm in these two cases is the same algorithm, called Gradient Descent-Ascent (GDA). To motivate GDA, suppose we are trying to learn $\theta^\star$. We could do gradient ascent on the function $\theta \mapsto \min_{\eta \in \mathrm{H}} \mathcal{L}(\theta, \eta)$. But then we would need to take the derivative in θ of this function. To see how to do this, suppose that $\mathcal{L}$ is strongly convex in η so that there is one minimizer $\eta^\star(\theta)$ of $\mathcal{L}(\theta, \cdot)$. Then, *Danskin's theorem* says that

$$\nabla_\theta \left[\min_{\eta \in \mathrm{H}} \mathcal{L}(\theta, \eta) \right] = \nabla_\theta \mathcal{L}(\theta, \texttt{stop\_grad}(\eta^\star(\theta))), \tag{A.3.23}$$

where the gradient is *only with respect to the first argument* (i.e., not a total derivative which would require computing the Jacobian of $\eta^\star(\theta)$ with respect to θ), indicated by the stop-gradient operator.[8] In order to take the derivative in θ, we need to set up a *secondary process* to also optimize η to obtain an approximation for $\eta^\star(\theta)$. We can do this through the following algorithmic template:

$$\eta_{k+1} = \eta_{k+1}^{T+1}; \qquad \eta_{k+1}^{t+1} = \eta_{k+1}^t - h \nabla_\eta \mathcal{L}(\theta_k, \eta_{k+1}^t), \quad \forall t \in [T]; \qquad \eta_{k+1}^1 = \eta_k \tag{A.3.24}$$

$$\theta_{k+1} = \theta_k + h \nabla_\theta \mathcal{L}(\theta_k, \eta_{k+1}). \tag{A.3.25}$$

That is, we take T steps of gradient descent to update η_k (hopefully to the minimizer of $\mathcal{L}(\theta_k, \cdot)$), and then take a gradient ascent step to update θ_k. As a bonus, on top of estimating $\theta_K \approx \arg\max_\theta \min_\eta \mathcal{L}(\theta, \eta)$, we also learn an $\eta_K \approx \arg\min_\eta \mathcal{L}(\theta_K, \eta)$ — this is an approximate Stackelberg equilibrium.

[8] In the case that $\mathcal{L}$ is not strongly convex in η but rather just convex, Danskin's theorem can be stated in terms of subdifferentials. If $\mathcal{L}$ is not convex at all in η, then this derivative may not be well-defined, but one can obtain (local) convergence guarantees for the resulting algorithm anyways. Hence we use Danskin's theorem as a motivation and not a justification for our algorithms. Danskin's theorems can be generalized into more relaxed circumstances by the so-called *envelope theorems*.

This method is often not done in practice, as it requires $T+1$ total gradient descent iterations to update θ once. Instead, we use the so-called *(simultaneous) Gradient Descent-Ascent* (GDA) iteration

$$\theta_{k+1} = \theta_k + h\nabla_\theta \mathcal{L}(\theta_k, \eta_k) \tag{A.3.26}$$

$$\eta_{k+1} = \eta_k - Th\nabla_\eta \mathcal{L}(\theta_k, \eta_k), \tag{A.3.27}$$

which can be implemented efficiently via a single gradient step on (θ, η). The crucial idea here is, to make our method close to the inefficient iteration above, we use an η update which is T times faster than the θ update (these can be seen as nearly the same by taking a linearization of the dynamics).

It is crucial to pick T sensibly. How can we do that? In the sequel, we discuss two configurations of T which lead to convergence of GDA to a Stackelberg equilibrium under different assumptions.

Convergence of One-Timescale GDA to Stackelberg Equilibrium

If $T = 1$ (i.e., named *one-timescale* because both θ and η updates are of the same scale), then the GDA algorithm becomes

$$\theta_{k+1} = \theta_k + h\nabla_\theta \mathcal{L}(\theta_k, \eta_k), \qquad \eta_{k+1} = \eta_k - h\nabla_\eta \mathcal{L}(\theta_k, \eta_k). \tag{A.3.28}$$

If $\mathcal{L}$ has Lipschitz gradients in θ and η, and is strongly concave in θ and strongly convex in η, and is coercive (i.e.,$\lim_{\|\theta\|_2, \|\eta\|_2 \to \infty} \mathcal{L}(\theta, \eta) = \infty$), then all saddle points are Stackelberg equilibria and vice versa. The work [ZAK24] shows that GDA (again, with $T = 1$) with sufficiently small step size h converges to a saddle point (hence Stackelberg equilibrium) exponentially fast if one exists, analogously to gradient descent for strongly convex functions with Lipschitz gradient.[9] To our knowledge, this flavor of results constitute the only known rigorous justification for single-timescale GDA.

Local Convergence of Two-Timescale GDA to Stackelberg Equilibrium

Strong convexity/concavity is a *global* property, and none of the games we look into in this book have objectives which are globally strongly concave/strongly convex. In this case, the best we can hope for is *local* convergence to Stackelberg equilibria: if the parameters are initialized close to a Stackelberg equilibrium, then GDA can converge onto it, given an appropriate step size h and timescale T.

In fact, our results also hold for a version of the *local* Stackelberg equilibrium called the *differential Stackelberg equilibrium*, which was introduced in [FCR19] (though we use the precise definition in [LFD+22]), and which we define as follows. A point $(\theta^\star, \eta^\star)$ is a differential Stackelberg equilibrium if:

[9] We do not provide the step size or exponential base since they are complicated functions of the strong convexity/concavity/Lipschitz constant of the gradient. Of course, the paper [ZAK24] provides the precise parameter values.

- $\nabla_\eta \mathcal{L}(\theta^\star, \eta^\star) = \mathbf{0}$;
- $\nabla^2_\eta \mathcal{L}(\theta^\star, \eta^\star)$ is symmetric positive definite;
- $\nabla_\theta \mathcal{L}(\theta^\star, \eta^\star) = \mathbf{0}$;
- $(\nabla^2_\theta \mathcal{L} + [\frac{\mathrm{d}}{\mathrm{d}\theta}\nabla_\eta \mathcal{L}][\nabla^2_\eta \mathcal{L}]^{-1}[\frac{\mathrm{d}}{\mathrm{d}\eta}\nabla_\theta \mathcal{L}])(\theta^\star, \eta^\star)$ is symmetric negative definite.

Notice that the last condition asks for the (total) Hessian

$$\nabla^2 \mathcal{L}(\theta, \eta) = \begin{bmatrix} \nabla^2_\theta \mathcal{L}(\theta, \eta) & \frac{\mathrm{d}}{\mathrm{d}\eta}\nabla^2_\theta \mathcal{L}(\theta, \eta) \\ \frac{\mathrm{d}}{\mathrm{d}\theta}\nabla^2_{\eta\theta} \mathcal{L}(\theta, \eta) & \nabla^2_\eta \mathcal{L}(\theta, \eta) \end{bmatrix} \tag{A.3.29}$$

or, equivalently, its Schur complement to be negative definite. If we look at the computation of $\nabla_\theta[\min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta)]$ furnished by Danskin's theorem, the last two criteria are actually constraints on the gradient and Hessian of the function $\theta \mapsto \min_{\eta\in\mathrm{H}} \mathcal{L}(\theta, \eta)$, ensuring that the gradient is 0 and the Hessian is negative semidefinite. This intuition tells us that we can expect that each Stackelberg equilibrium is a differential Stackelberg equilibrium; [FCR20] confirms this rigorously (up to some technical conditions).

Analogously to the notion of strict local optimum in single-player optimization (where we require $\nabla^2 \mathcal{L}(\theta^\star)$ to be positive semidefinite), the definition of differential Stackelberg equilibrium *implies that* $\mathcal{L}(\theta, \cdot)$ *is locally (strictly) convex in a neighborhood of the equilibrium, and that* $\min_{\eta\in\mathrm{H}} \mathcal{L}(\cdot, \eta)$ *is locally (strictly) concave in the same region.*

In this context, we present the result from [LFD+22]. Let $(\theta^\star, \eta^\star)$ be a differential Stackelberg equilibrium. Suppose that $\mathcal{L}$ has Lipschitz gradients, i.e.,

$$\max\left\{ \|\nabla^2_\theta \mathcal{L}(\theta^\star, \eta^\star)\|_2, \left\|\nabla^2_\eta \mathcal{L}(\theta^\star, \eta^\star)\right\|_2, \left\| \frac{\mathrm{d}}{\mathrm{d}\eta}\nabla_\theta \mathcal{L}(\theta^\star, \eta^\star) \right\|_2 \right\} \leq \beta. \tag{A.3.30}$$

Further define the local strong convexity/concavity parameters of $\mathcal{L}(\theta, \cdot)$ and $\min_\eta \mathcal{L}(\cdot, \eta)$ respectively as

$$\mu_\eta = \lambda_{\min}(\nabla^2_\eta \mathcal{L}(\theta^\star, \eta^\star)), \tag{A.3.31}$$

$$\mu_\theta = \min\left\{ \beta, -\left(\nabla^2_\theta \mathcal{L} + \left[\frac{\mathrm{d}}{\mathrm{d}\theta}\nabla_\eta \mathcal{L}\right] \left[\nabla^2_\eta \mathcal{L}\right]^{-1} \left[\frac{\mathrm{d}}{\mathrm{d}\eta}\nabla_\theta \mathcal{L}\right] \right)(\theta^\star, \eta^\star) \right\}. \tag{A.3.32}$$

Then define the local condition numbers as

$$\kappa_\eta = \beta/\mu_\eta, \qquad \kappa_\theta = \beta/\mu_\theta. \tag{A.3.33}$$

The paper [LFD+22] says that if we take step size $h = \frac{1}{4\beta T}$ and take $T \geq 2\kappa_\eta$, so that the algorithm is

$$\theta_{k+1} = \theta_k + \frac{1}{4\beta T}\nabla_\theta \mathcal{L}(\theta_k, \eta_k), \tag{A.3.34}$$

$$\eta_{k+1} = \eta_k - \frac{1}{4\beta}\nabla_\eta \mathcal{L}(\theta_k, \eta_k), \tag{A.3.35}$$

the total Hessian $\nabla^2\mathcal{L}(\theta^\star, \eta^\star)$ is diagonalizable, and we initialize (θ_0, η_0) close enough to $(\theta^\star, \eta^\star)$, then there are positive constants $c_0, c_1 > 0$ such that

$$\|(\theta_k, \eta_k) - (\theta^\star, \eta^\star)\|_2 \leq c_0 \left(1 - \frac{c_1}{T\kappa_\theta}\right)^k \|(\theta_0, \eta_0) - (\theta^\star, \eta^\star)\|_2, \tag{A.3.36}$$

implying exponential convergence to the differential Stackelberg equilibrium.

A.3.2 Practical Considerations when Learning Stackelberg Equilibria

In practice, we do not know how to initialize parameters close to a (differential) Stackelberg equilibrium. Due to symmetries within the objective, including those induced by overparameterization of the neural networks being trained, one can (heuristically) expect that most initializations are close to a Stackelberg equilibrium. Also, we do not know how to compute the step size h or the timescale T, since they are dependent on properties of the loss $\mathcal{L}$ at the equilibrium. In practice, there are some common approaches:

- Take $T = 1$ (equal step-sizes), and use updates for θ and η that are derived from a learning-rate-adaptive optimizer like Adam (as opposed to vanilla GD). Here, you *hope* (but do not *know*) that the optimizer can adjust the learning rates to learn a good equilibrium.
- Take T to be some constant like $T = 10^6$ which implies that η equilibrates 10^6 times as fast as θ. Here you can also use Adam-style updates, and hope that it fixes the time scale.
- Let T depend on the iteration k, and let $T_k \to \infty$ as $k \to \infty$. This schedule was studied (also in the case of noise) by Borkar in [Bor97].

For example, you can use this while training CTRL-style models (see Chapter 6), where the encoder is Player 1 and the decoder is Player 2. Some theory about CTRL is given in Theorem 6.1.

A.4 Exercises

Exercise A.1. We have shown that for a smooth function f, gradient descent converges linearly to the global optimum if it is strongly convex. However, in general nonconvex optimization, we do not have convexity, let alone strong convexity. Fortunately, in some cases, f satisfies the so-called μ-Polyak-Lojasiewicz (PL) inequality, i.e., there exists a constant $\mu > 0$ such that for all θ,

$$\frac{1}{2}\|\nabla f(\theta)\|_2^2 \geq \mu\left(f(\theta) - f(\theta^\star)\right),$$

where θ^* is a minimizer of f.

Please show that under the PL inequality and the assumption that f is β-smooth, gradient descent (A.1.12) converges linearly to θ^*.

Exercise A.2. Compute the differential and adjoint derivative of the softmax function, defined as follows.

$$\operatorname{softmax}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \frac{1}{\sum_{i=1}^n e^{x_i}} \begin{bmatrix} e^{x_1} \\ \vdots \\ e^{x_n} \end{bmatrix}. \tag{A.4.1}$$

Exercise A.3. Carry through the backpropagation computation for a L-layer MLP, as defined in Section 8.2.3.

Exercise A.4. Carry through the backpropagation computation for a L-layer transformer, as defined in Section 8.2.3.

Exercise A.5. Carry through the backpropagation computation for an autoencoder with L encoder layers and L decoder layers (without necessarily specifying an architecture).

Appendix B

Entropy, Diffusion, Denoising, and Lossy Coding

> "*The increase of disorder or entropy with time is one example of what is called an arrow of time, something that distinguishes the past from the future, giving a direction to time.*"
>
> – A Brief History of Time, Stephen Hawking

In this appendix we provide proofs for several facts, mentioned in Chapter 3, which are related to differential entropy, how it evolves under diffusion processes, and its connections to lossy coding. We will make the following mild assumption about the random variable representing the data source, denoted $\boldsymbol{x}$.

Assumption B.1. $\boldsymbol{x}$ is supported on a compact set $\mathcal{S} \subseteq \mathbb{R}^D$ of radius at most R, i.e., $R \doteq \sup_{\boldsymbol{\xi} \in \mathcal{S}} \|\boldsymbol{\xi}\|_2$.

In particular, since compact sets in Euclidean space are bounded, it holds $R < \infty$. We will consistently use the notation $B_r(\boldsymbol{\xi}) \doteq \{\boldsymbol{u} \in \mathbb{R}^D : \|\boldsymbol{\xi} - \boldsymbol{u}\|_2 \leq r\}$ to denote the Euclidean ball of radius r centered at $\boldsymbol{\xi}$. In this sense, Assumption B.1 has $\mathcal{S} \subseteq B_R(\mathbf{0})$.

Notice that this assumption holds for (almost) all variables we care about in practice, as it is (often) imposed by a normalization step during data preprocessing.

B.1 Differential Entropy of Low-Dimensional Distributions

In this short appendix we discuss the differential entropy of low-dimensional distributions. By definition, the differential entropy of a random variable $\boldsymbol{x}$ which does not have a density is $-\infty$; this includes all random variables supported on low-dimensional sets. The objective of this section is to discuss why this is a "morally correct" value.

In fact, let $\boldsymbol{x}$ be any random variable such that Assumption B.1 holds, the support $\mathcal{S}$ of $\boldsymbol{x}$ has 0 volume.[1] We will consider the case that $\boldsymbol{x}$ is uniform on $\mathcal{S}$.[2] Our goal is to compute $h(\boldsymbol{x})$.

In this case, $\boldsymbol{x}$ would not have a density; in the counterfactual world where we did not know $h(\boldsymbol{x}) = -\infty$, we could not directly define it using the standard definition of differential entropy. Instead, as in the rest of analysis and information theory it would be reasonable to consider the limit of entropies of successively better approximations $\boldsymbol{x}_\varepsilon$ of $\boldsymbol{x}$ which have densities, i.e.,

$$h(\boldsymbol{x}) \text{ “=” } \lim_{\varepsilon \searrow 0} h(\boldsymbol{x}_\varepsilon). \tag{B.1.1}$$

To this end, the basic idea is to take an ε-thickening of $\mathcal{S}$, say $\mathcal{S}_\varepsilon$ defined as

$$\mathcal{S}_\varepsilon = \bigcup_{\boldsymbol{\xi} \in \mathcal{S}} B_\varepsilon(\boldsymbol{\xi}) \tag{B.1.2}$$

and visualized in Figure B.1. We will work with random variables whose support

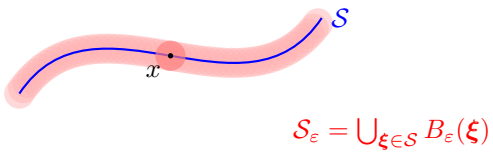

Figure B.1: Illustration of the ε-thickening $\mathcal{S}_\varepsilon$ of a curve $\mathcal{S} \subseteq \mathbb{R}^2$.

is $\mathcal{S}_\varepsilon$, which is fully-dimensional, and take the limit as $\varepsilon \to 0$. Indeed, define $\boldsymbol{x}_\varepsilon \sim \mathcal{U}(\mathcal{S}_\varepsilon)$. Since $\mathcal{S}_\varepsilon$ has positive volume, $\boldsymbol{x}_\varepsilon$ has a density p_ε equal to

$$p_\varepsilon(\boldsymbol{\xi}) = \mathbf{1}(\boldsymbol{\xi} \in \mathcal{S}_\varepsilon) \cdot \frac{1}{\mathrm{vol}(\mathcal{S}_\varepsilon)}. \tag{B.1.3}$$

Computing the entropy of $\boldsymbol{x}_\varepsilon$ using the convention that $0 \log 0 = 0$, it holds

$$h(\boldsymbol{x}_\varepsilon) = -\int_{\mathbb{R}^D} p_\varepsilon(\boldsymbol{\xi}) \log p_\varepsilon(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \tag{B.1.4}$$

[1] Formally this means that $\mathcal{S}$ is Borel measurable with Borel measure 0.
[2] Say, w.r.t. the Hausdorff measure on $\mathcal{S}$.

$$= -\int_{\mathcal{S}_\varepsilon} \frac{1}{\operatorname{vol}(\mathcal{S}_\varepsilon)} \log\left(\frac{1}{\operatorname{vol}(\mathcal{S}_\varepsilon)}\right) \mathrm{d}\boldsymbol{\xi} \tag{B.1.5}$$

$$= \frac{\log(\operatorname{vol}(\mathcal{S}_\varepsilon))}{\operatorname{vol}(\mathcal{S}_\varepsilon)} \int_{\mathcal{S}_\varepsilon} \mathrm{d}\boldsymbol{\xi} \tag{B.1.6}$$

$$= \log(\operatorname{vol}(\mathcal{S}_\varepsilon)). \tag{B.1.7}$$

Since $\mathcal{S}$ is compact $\operatorname{vol}(\mathcal{S}_\varepsilon)$ is finite and tends to 0 as $\varepsilon \searrow 0$. Thus

$$h(\boldsymbol{x}) = \lim_{\varepsilon \searrow 0} h(\boldsymbol{x}_\varepsilon) = \lim_{\varepsilon \searrow 0} \log(\operatorname{vol}(\mathcal{S}_\varepsilon)) = -\infty, \tag{B.1.8}$$

as desired.

The above calculation is actually a corollary of a much more famous and celebrated set of results about the maximum possible entropy of $\boldsymbol{x}$ subject to certain constraints on the distribution of $\boldsymbol{x}$. We would be remiss to not provide the results here; the proofs are provided in Chapter 2 of [PW22], for example.

Theorem B.1. *Let $\boldsymbol{x}$ be a random variable on $\mathbb{R}^D$.*

1. *If $\boldsymbol{x}$ is supported on a compact set $\mathcal{S} \subseteq \mathbb{R}^D$ (i.e., Assumption B.1) then*

$$h(\boldsymbol{x}) \leq h(\mathcal{U}(\mathcal{S})) = \log \operatorname{vol}(\mathcal{S}). \tag{B.1.9}$$

2. *If $\boldsymbol{x}$ has finite covariance such that, for a PSD matrix $\boldsymbol{\Sigma} \in \mathsf{PSD}(D)$, it holds $\operatorname{Cov}(\boldsymbol{x}) \preceq \boldsymbol{\Sigma}$ (w.r.t. the PSD ordering, i.e., $\boldsymbol{\Sigma} - \operatorname{Cov}(\boldsymbol{x})$ is PSD), then*

$$h(\boldsymbol{x}) \leq h(\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})) = \frac{1}{2} \log((2\pi e)^D \det \boldsymbol{\Sigma}). \tag{B.1.10}$$

3. *If $\boldsymbol{x}$ has finite second moment such that, for a constant $a \geq 0$, it holds $\mathbb{E} \|\boldsymbol{x}\|_2^2 \leq a$, then*

$$h(\boldsymbol{x}) \leq h\left(\mathcal{N}\left(\mathbf{0}, \frac{a}{D}\boldsymbol{I}\right)\right) = \frac{D}{2} \log \frac{2\pi e a}{D}. \tag{B.1.11}$$

B.2 Diffusion and Denoising Processes

In the main body (Chapter 3), we considered a random variable $\boldsymbol{x}$, and a stochastic process defined by (3.2.1), i.e.,

$$\boldsymbol{x}_t = \boldsymbol{x} + t\boldsymbol{g}, \qquad \forall t \in [0, T] \tag{B.2.1}$$

where $\boldsymbol{g} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ independently of $\boldsymbol{x}$.

The structure of this section is as follows. In Section B.2.1 we provide a formal theorem and crisp proof which shows that under Equation (B.2.1) the entropy increases, i.e., $\frac{\mathrm{d}}{\mathrm{d}t} h(\boldsymbol{x}_t) > 0$. In Section B.2.2 we provide a formal theorem and crisp proof which shows that under Equation (B.2.1), the entropy

decreases during denoising, i.e., $h(\mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t]) < h(\boldsymbol{x}_t)$ for all $s < t$. In Section B.2.3 we provide proofs for technical lemmas that are needed to establish the claims in the previous subsections.

Before we start, we introduce some key notations. First, let φ_t be the density of $\mathcal{N}(\mathbf{0}, t^2\boldsymbol{I})$, i.e.,

$$\varphi_t(\boldsymbol{\xi}) \doteq \frac{1}{(2\pi)^{D/2}t^D} \exp\left(-\frac{\|\boldsymbol{\xi}\|_2^2}{2t^2}\right). \tag{B.2.2}$$

Next, $\boldsymbol{x}_t$ is supported on all of $\mathbb{R}^D$, so it has a *density*, which we denote p_t (as in the main body). A quick calculation shows that

$$p_t(\boldsymbol{\xi}) = \mathbb{E}[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x})], \tag{B.2.3}$$

and from this representation it is possible to deduce (i.e., from Proposition B.4) that p_t is smooth (i.e., infinitely differentiable) in $\boldsymbol{\xi}$, also smooth in t, and positive everywhere. This fact is somewhat remarkable at first sight: even for a completely irregular random variable $\boldsymbol{x}$ (say, a Bernoulli random variable, which does not have a density), its Gaussian smoothing admits a density for every (arbitrarily small) $t > 0$. The proof is left as an exercise for readers well-versed in mathematical analysis.

However, we also need to add an assumption about the *smoothness* of the distribution of $\boldsymbol{x}$, which will eliminate some technical problems that occur around $t = 0$ with low-dimensional distributions.[3] Despite this, we expect that our results hold under milder assumptions with additional work. For now, let us assume:

Assumption B.2. $\boldsymbol{x}$ has a twice continuously differentiable density, denoted p.

B.2.1 Diffusion Process Increases Entropy Over Time

In this section we provide a proof of Theorem B.2. For convenience, we restate it as follows.

Theorem B.2 (Diffusion Increases Entropy)**.** *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). Then*

$$h(\boldsymbol{x}_s) < h(\boldsymbol{x}_t), \qquad \forall s, t\colon 0 \le s < t \le T. \tag{B.2.4}$$

Proof. Before we start, we must ask: when does the inequality in (B.2.4) make sense? We will show in Lemma B.1 that under our assumptions, the differential entropy is well-defined, is never $+\infty$, and for $t > 0$ is finite, so the (strict) inequality in (B.2.4) makes sense.

The question of well-definedness aside, the crux of this proof is to show that the density p_t of $\boldsymbol{x}_t$ satisfies a particular partial differential equation, which is

[3] As then various quantities become highly irregular and dealing with them would require significant additional analysis.

very similar to the *heat equation*. The heat equation is a famous PDE which describes the diffusion of heat through space. This intuitively should make sense, and paints a mental picture: as the time t increases, the probability from the original (perhaps tightly concentrated) $\boldsymbol{x}$ disperses across all of $\mathbb{R}^D$ like heat radiating from a source in a vacuum.

Such PDEs for p_t, known as *Fokker-Planck equations* for more general stochastic processes, are very powerful tools, as they allow us to describe the instantaneous temporal derivatives of p_t in terms of the instantaneous spatial derivatives of p_t, and vice versa, providing a concise description of the regularity and dynamics of p_t. Once we obtain dynamics for p_t, we can then use the system to obtain another one which describes the dynamics of $h(\boldsymbol{x}_t)$, which after all is just a functional of p_t.

The description of the PDE involves a mathematical object called the Laplacian Δ. Recall from your multivariable calculus class that the Laplacian operating on a differentiable-in-time and twice-differentiable-in-space function $f\colon (0,T)\times \mathbb{R}^D \to \mathbb{R}$ is given by

$$\Delta f_t(\boldsymbol{\xi}) = \operatorname{tr}(\nabla^2 f_t(\boldsymbol{\xi})) = \sum_{i=1}^{D} \frac{\partial^2 f_t}{\partial \xi_i^2}(\boldsymbol{\xi}). \tag{B.2.5}$$

Namely, from using the integral representation of p_t and differentiating under the integral, we can compute the derivatives of p_t (which we do in Proposition B.1) and observe that p_t satisfies the heat-like PDE

$$\frac{\partial p_t}{\partial t}(\boldsymbol{\xi}) = t\Delta p_t(\boldsymbol{\xi}). \tag{B.2.6}$$

Then for finding the dynamics of $h(\boldsymbol{x}_t)$, we can use Proposition B.3 again as well as the heat-like PDE to get

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} h(\boldsymbol{x}_t) &= -\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} && \text{(B.2.7)}\\
&= -\int_{\mathbb{R}^D} \frac{\partial}{\partial t} \left[p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi})\right] \mathrm{d}\boldsymbol{\xi} && \text{(B.2.8)}\\
&= -\int_{\mathbb{R}^D} \frac{\partial p_t}{\partial t}(\boldsymbol{\xi}) [1 + \log p_t(\boldsymbol{\xi})] \mathrm{d}\boldsymbol{\xi} && \text{(B.2.9)}\\
&= -t \int_{\mathbb{R}^D} \Delta p_t(\boldsymbol{\xi}) [1 + \log p_t(\boldsymbol{\xi})] \mathrm{d}\boldsymbol{\xi}. && \text{(B.2.10)}
\end{aligned}$$

By using a slightly involved integration by parts argument (Lemma B.2), we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} h(\boldsymbol{x}_t) &= t \int_{\mathbb{R}^D} \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi}) \rangle \mathrm{d}\boldsymbol{\xi} && \text{(B.2.11)}\\
&= t \int_{\mathbb{R}^D} \frac{\|\nabla p_t(\boldsymbol{\xi})\|_2^2}{p_t(\boldsymbol{\xi})} \mathrm{d}\boldsymbol{\xi} && \text{(B.2.12)}
\end{aligned}$$

$$> 0 \tag{B.2.13}$$

where strict inequality holds in the last line because, for it to not hold, $\nabla p_t(\boldsymbol{\xi})$ would need to vanish almost everywhere (i.e., everywhere except possibly on a set of zero volume), but this would imply that p_t would be constant almost everywhere, a contradiction with a fact that p_t is a density.

To complete the proof we just use the fundamental theorem of calculus

$$h(\boldsymbol{x}_t) = h(\boldsymbol{x}_s) + \int_s^t \frac{\mathrm{d}}{\mathrm{d}u} h(\boldsymbol{x}_u)\mathrm{d}u > h(\boldsymbol{x}_s), \tag{B.2.14}$$

which proves the claim. (Note that this does not make sense when $h(\boldsymbol{x}_s) = -\infty$, which can only happen when $s = 0$ and $h(\boldsymbol{x}) = -\infty$, but in this case $h(\boldsymbol{x}_t) > -\infty$ so the claim is vacuously true anyways.) □

B.2.2 Denoising Process Reduces Entropy Over Time

Recall that in Section 3.2.1 we start with the random variable $\boldsymbol{x}_T$ and iteratively denoise it using iterations of the form

$$\hat{\boldsymbol{x}}_s \doteq \mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t = \hat{\boldsymbol{x}}_t] = \frac{s}{t}\hat{\boldsymbol{x}}_t + \left(1 - \frac{s}{t}\right)\bar{\boldsymbol{x}}^*(t, \hat{\boldsymbol{x}}_t). \tag{B.2.15}$$

for $s, t \in \{t_0, t_1, \ldots, t_L\}$ with $s < t$ and $\boldsymbol{x}_T = \hat{\boldsymbol{x}}_T$. We wish to prove that $h(\hat{\boldsymbol{x}}_s) < h(\hat{\boldsymbol{x}}_t)$, showing that the denoising process actually reduces the entropy.

Before we go about doing this, we make several remarks about the problem statement. First, Tweedie's formula (3.2.23) says that

$$\bar{\boldsymbol{x}}^*(t, \boldsymbol{x}_t) = \boldsymbol{x}_t + t^2 \nabla p_t(\boldsymbol{x}_t), \tag{B.2.16}$$

which likens a full denoising step from time t to time 0 to a gradient step on the log-density of $\boldsymbol{x}_t$. Can we get a similar result for the full denoising step from time t to time s in (B.2.15)? It turns out that indeed we can, and it is pretty simple. By using (B.2.15) and Tweedie's formula (3.2.23), we obtain

$$\mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t] = \frac{s}{t}\boldsymbol{x}_t + \left(1 - \frac{s}{t}\right)\left(\boldsymbol{x}_t + t^2 \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right) = \boldsymbol{x}_t + \left(1 - \frac{s}{t}\right) t^2 \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t). \tag{B.2.17}$$

So this iterative denoising step is again a gradient step on the perturbed log-density $\log p_t$ with a shrunken step size. In particular, this step can be seen as a perturbation of the distribution of the random variable $\boldsymbol{x}_t$ by the *score function vector field*, suggesting a connection to stochastic differential equations (SDEs) and the theory of diffusion models [SSK+21]. Indeed, a proof of the following result Theorem B.3 can be developed using this powerful machinery and a limiting argument (e.g., following the technical approach in the exposition of [CCL+23]). We will give a simpler proof here, which will use only elementary tools and thereby illuminate some of the key quantities behind the process of entropy reduction via denoising. On the other hand, we will need to deal with some slightly technical calculations due to the fact that the denoising process

in Theorem B.3 does *not* correspond exactly to the *reverse* process associated to the noise addition process that generates the observation $\boldsymbol{x}_t$.[4]

We want to prove that $h(\mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t]) < h(\boldsymbol{x}_t)$, i.e., formally:

Theorem B.3. *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). For each $t > 0$, write*

$$J(p_t) \doteq \int_{\mathbb{R}^D} \frac{\|\nabla p_t(\boldsymbol{\xi})\|_2^2}{p_t(\boldsymbol{\xi})} \mathrm{d}\boldsymbol{\xi}, \qquad U_t \doteq \max\left(\frac{D}{t^2}, \left|\frac{R^2}{t^4} - \frac{D}{t^2}\right|\right) \tag{B.2.18}$$

for the Fisher information of p_t and a uniform bound on $|\Delta \log p_t|$, respectively. Then

$$h(\mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t]) < h(\boldsymbol{x}_t) \tag{B.2.19}$$

for all $0 \le s < t \le T$ such that

$$\left(1 - \frac{s}{t}\right) t^2 U_t^2 \exp\left(\left(1 - \frac{s}{t}\right) t^2 U_t\right) < 2J(p_t). \tag{B.2.20}$$

Proof. This proof uses two main ideas:

1. First, write down a density for $\mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t]$ using a change-of-variables formula.

2. Second, bound this density to control the entropy.

The change of variables is justified by Corollary B.1, which was originally derived in [Gri11].

We execute these ideas now. From Corollary B.1, we obtain that the function $\bar{\boldsymbol{x}}$ defined as $\bar{\boldsymbol{x}}(\boldsymbol{\xi}) \doteq \mathbb{E}[\boldsymbol{x}_s \mid \boldsymbol{x}_t = \boldsymbol{\xi}]$ is differentiable, injective, and thus invertible on its range, which we henceforth denote $\mathcal{X} \subseteq \mathbb{R}^D$. We denote its inverse as $\bar{\boldsymbol{x}}^{-1}$. Using a change-of-variables formula, the density $\bar{p}$ of $\bar{\boldsymbol{x}}(\boldsymbol{x}_t)$ is given by

$$\bar{p}(\boldsymbol{\xi}) \doteq \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))}, \tag{B.2.21}$$

where (recall, from Section A.2) $\bar{\boldsymbol{x}}'$ is the Jacobian of $\bar{\boldsymbol{x}}$. Since from Lemma B.3 we know $\bar{\boldsymbol{x}}'$ is a positive definite matrix, the determinant is positive and so the whole density is positive. Then it follows that

$$h(\bar{\boldsymbol{x}}(\boldsymbol{x}_t)) = -\int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \mathrm{d}\boldsymbol{\xi} \tag{B.2.22}$$

$$= -\int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log((p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})) \mathrm{d}\boldsymbol{\xi} \tag{B.2.23}$$

[4] For those familiar with diffusion models, we refer here to the time-reversed forward process not coinciding with the sequence of iterates generated by the process defined by Theorem B.3. These processes coincide in a certain limit where infinitely many steps of Theorem B.3 are taken with infinitely small levels of noise added at each step; for general, finite steps, we must introduce some approximations regardless of the level of sophistication of our tools.

$$+\int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log\det\big(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi}))\big)\mathrm{d}\boldsymbol{\xi} \tag{B.2.24}$$

$$= -\int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} + \int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log\det\big(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi}))\big)\mathrm{d}\boldsymbol{\xi} \tag{B.2.25}$$

$$= h(\boldsymbol{x}_t) - \int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log\left(\frac{1}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))}\right)\mathrm{d}\boldsymbol{\xi}. \tag{B.2.26}$$

We will study the last term (including the $-$), and show that it is negative.

By concavity, one has $-x \log x \leq 1 - x$ for every $x \geq 0$. Hence

$$h(\bar{\boldsymbol{x}}(\boldsymbol{x}_t)) - h(\boldsymbol{x}_t) = -\int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))} \log\left(\frac{1}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))}\right)\mathrm{d}\boldsymbol{\xi} \tag{B.2.27}$$

$$\leq \int_{\mathcal{X}} (p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi}) \cdot \left(1 - \frac{1}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))}\right)\mathrm{d}\boldsymbol{\xi} \tag{B.2.28}$$

$$= \int_{\mathcal{X}} (p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} - \int_{\mathcal{X}} \frac{(p_t \circ \bar{\boldsymbol{x}}^{-1})(\boldsymbol{\xi})}{\det(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi})))}\mathrm{d}\boldsymbol{\xi} \tag{B.2.29}$$

$$= \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \det\big(\bar{\boldsymbol{x}}'(\bar{\boldsymbol{x}}^{-1}(\boldsymbol{\xi}))\big)\mathrm{d}\boldsymbol{\xi} - \int_{\mathcal{X}} \bar{p}(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \tag{B.2.30}$$

$$= \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \det\Big(\boldsymbol{I} + \Big(1 - \frac{s}{t}\Big)\, t^2 \nabla^2 \log p_t(\boldsymbol{\xi})\Big)\mathrm{d}\boldsymbol{\xi} - 1. \tag{B.2.31}$$

Now, by the AM-GM inequality on eigenvalues, we have for any symmetric positive definite matrix $\boldsymbol{M} \in \mathsf{PSD}(D)$ the bound

$$\det(\boldsymbol{M})^{1/D} = \prod_{i=1}^{D} \lambda_i(\boldsymbol{M})^{1/D} \leq \frac{\sum_{i=1}^{D} \lambda_i(\boldsymbol{M})}{D} = \frac{\mathrm{tr}(\boldsymbol{M})}{D}, \tag{B.2.32}$$

which we can apply to the above expression and obtain

$$\int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \det\Big(\boldsymbol{I} + \Big(1 - \frac{s}{t}\Big)\, t^2 \nabla^2 \log p_t(\boldsymbol{\xi})\Big)\mathrm{d}\boldsymbol{\xi} \tag{B.2.33}$$

$$\leq \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \,\mathrm{tr}\left(\frac{1}{D}\left[\boldsymbol{I} + \Big(1 - \frac{s}{t}\Big)\, t^2 \nabla^2 \log p_t(\boldsymbol{\xi})\right]\right)^D \mathrm{d}\boldsymbol{\xi} \tag{B.2.34}$$

$$= \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \left(1 + \frac{\big(1 - \frac{s}{t}\big)\, t^2}{D} \,\mathrm{tr}(\nabla^2 \log p_t(\boldsymbol{\xi}))\right)^D \mathrm{d}\boldsymbol{\xi} \tag{B.2.35}$$

$$= \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \left(1 + \frac{\big(1 - \frac{s}{t}\big)\, t^2}{D} \Delta \log p_t(\boldsymbol{\xi})\right)^D \mathrm{d}\boldsymbol{\xi}. \tag{B.2.36}$$

Since $\bar{\boldsymbol{x}}'(\boldsymbol{\xi})$ is positive definite (Corollary B.1), we have $1 + \frac{(1-s/t)t^2}{D} \Delta \log p_t(\boldsymbol{\xi}) =$

$\frac{\operatorname{tr}(\bar{\boldsymbol{x}}'(\boldsymbol{\xi}))}{D} > 0$. By the inequality $\log(1+x) \le x$ for $x > -1$,

$$\left(1 + \frac{\left(1-\frac{s}{t}\right)t^2}{D}\Delta \log p_t(\boldsymbol{\xi})\right)^D \le \exp\left(\left(1-\frac{s}{t}\right)t^2 \Delta \log p_t(\boldsymbol{\xi})\right). \tag{B.2.37}$$

From Lemma B.5 (where, recall, R is the radius of the support of $\boldsymbol{x}$ as in Assumption B.1),

$$|\Delta \log p_t(\boldsymbol{\xi})| \le \max\left(\frac{D}{t^2}, \left|\frac{R^2}{t^4} - \frac{D}{t^2}\right|\right) =: U_t. \tag{B.2.38}$$

Setting $\varepsilon \doteq \left(1-\frac{s}{t}\right)t^2 U_t$, we have $\left(1-\frac{s}{t}\right)t^2 \Delta \log p_t(\boldsymbol{\xi}) \le \varepsilon$ for all $\boldsymbol{\xi}$. By Taylor's theorem with Lagrange remainder, for any $c \le \varepsilon$,

$$e^c \le 1 + c + \frac{c^2}{2}e^{\varepsilon}, \tag{B.2.39}$$

since $e^c = 1 + c + \frac{c^2}{2}e^{\theta c}$ for some $\theta \in (0,1)$, and $e^{\theta c} \le e^{\max(c,0)} \le e^{\varepsilon}$. Applying these bounds and integrating,

$$\int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \left(1 + \frac{\left(1-\frac{s}{t}\right)t^2}{D}\Delta \log p_t(\boldsymbol{\xi})\right)^D \mathrm{d}\boldsymbol{\xi} \tag{B.2.40}$$

$$\le \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \exp\left(\left(1-\frac{s}{t}\right)t^2 \Delta \log p_t(\boldsymbol{\xi})\right) \mathrm{d}\boldsymbol{\xi} \tag{B.2.41}$$

$$\le 1 + \left(1-\frac{s}{t}\right)t^2 \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \Delta \log p_t(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} + \frac{e^{\varepsilon}}{2}\left(1-\frac{s}{t}\right)^2 t^4 \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \left(\Delta \log p_t(\boldsymbol{\xi})\right)^2 \mathrm{d}\boldsymbol{\xi} \tag{B.2.42}$$

$$= 1 - \left(1-\frac{s}{t}\right)t^2 \int_{\mathbb{R}^D} \frac{\|\nabla p_t(\boldsymbol{\xi})\|_2^2}{p_t(\boldsymbol{\xi})} \mathrm{d}\boldsymbol{\xi} + \frac{e^{\varepsilon}}{2}\left(1-\frac{s}{t}\right)^2 t^4 \int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \left(\Delta \log p_t(\boldsymbol{\xi})\right)^2 \mathrm{d}\boldsymbol{\xi}, \tag{B.2.43}$$

where the identity $\int p_t \Delta \log p_t = -\int \|\nabla p_t\|_2^2 / p_t$ is the same as in the proof of Theorem B.2. Writing $J(p_t) \doteq \int_{\mathbb{R}^D} \frac{\|\nabla p_t(\boldsymbol{\xi})\|_2^2}{p_t(\boldsymbol{\xi})} \mathrm{d}\boldsymbol{\xi}$ for the Fisher information of p_t, and bounding $\left(\Delta \log p_t\right)^2 \le U_t^2$, we combine with our previous estimate to obtain

$$h(\bar{\boldsymbol{x}}(\boldsymbol{x}_t)) - h(\boldsymbol{x}_t) \le -\left(1-\frac{s}{t}\right)t^2 J(p_t) + \frac{e^{\varepsilon}\varepsilon^2}{2}. \tag{B.2.44}$$

This is strictly negative whenever $\varepsilon^2 e^{\varepsilon} < 2\left(1-\frac{s}{t}\right)t^2 J(p_t)$. Substituting $\varepsilon = \left(1-\frac{s}{t}\right)t^2 U_t$, this is precisely condition (B.2.20). □

Notice that condition (B.2.20) is always satisfiable: for any fixed $t > 0$, the Fisher information satisfies $J(p_t) > 0$ (since p_t is not constant) and $U_t < \infty$, so taking $\left(1-\frac{s}{t}\right)t^2$ small enough ensures (B.2.20) holds.

To understand the quantitative implications, consider the behavior as $t \to 0$ (near the data), where the condition is most restrictive. Setting $s = (1-\varepsilon)t$, the

condition becomes $\varepsilon t^2 U_t^2 \exp(\varepsilon t^2 U_t) < 2J(p_t)$. For small t, the Laplacian bound satisfies $U_t \approx R^2/t^4$, so the polynomial prefactor scales as $\varepsilon t^2 U_t^2 \approx \varepsilon R^4/t^6$. The critical scaling is therefore $\varepsilon \sim t^6/R^4$. Taking $\varepsilon = \alpha t^6/R^4$, the condition reduces to $\alpha < 2J(p_t)$. In other words, each denoising step of size $t - s = \varepsilon\, t \sim t^7/R^4$ reduces the entropy. The steps must become increasingly fine as $t \to 0$, a consequence of the second-order remainder in the Taylor expansion used in the proof.

At the same time, it should be noted that our bounds are likely loose in terms of the precise t-dependence, and certainly so for data with greater structure. For instance, the $U_t \sim R^2/t^4$ blowup reflects a worst-case Laplacian bound that can be considerably tightened under additional regularity assumptions on p. Moreover, if p_t is log-concave (as holds, e.g., when p itself is log-concave), then $\Delta \log p_t \leq 0$ everywhere, and a significantly simplified argument can be run with strengthened conclusions.

B.2.3 Technical Lemmas and Intermediate Results

In this subsection we present technical results which power our main two conceptual theorems. Our presentation will be more or less standard for mathematics; we will start with the higher-level results first, and gradually move back to the more incremental results. The higher-level results will use the incremental results, and in this way we have an easy-to-read dependency ordering of the results: no result depends on those before it. Results which do not depend on each other are generally ordered by the place they appear in the above pair of proofs.

Finiteness of the Differential Entropy

We first show that the entropy exists along the stochastic process and is finite.

Lemma B.1. *Let $\boldsymbol{x}$ be any random variable, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process* (B.2.1)*.*

1. *For $t > 0$, the differential entropy $h(\boldsymbol{x}_t)$ exists and is $> -\infty$.*

2. *If in addition Assumption B.1 holds for $\boldsymbol{x}$, then $h(\boldsymbol{x}) < \infty$ and $h(\boldsymbol{x}_t) < \infty$.*

Proof. To prove Lemma B.1.1, we use a classic yet tedious analysis argument. Since $\boldsymbol{x}_t$ has a density, we can write

$$h(\boldsymbol{x}_t) = -\int_{\mathbb{R}^D} p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}. \tag{B.2.45}$$

Accordingly, let $g\colon \mathbb{R}^D \to \mathbb{R}$ be defined as

$$g(\boldsymbol{\xi}) \doteq -p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}) \implies h(\boldsymbol{x}_t) = \int_{\mathbb{R}^D} g(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}. \tag{B.2.46}$$

As usual to bound the value of an integral in analysis, define

$$g_+(\boldsymbol{\xi}) \doteq \max(g(\boldsymbol{\xi}), 0), \quad g_-(\boldsymbol{\xi}) \doteq \max(-g(\boldsymbol{\xi}), 0) \tag{B.2.47}$$

$$\implies \quad g = g_+ - g_- \quad \text{and} \quad g_+, g_- \geq 0. \tag{B.2.48}$$

Then

$$h(\boldsymbol{x}_t) = \int_{\mathbb{R}^D} g_+(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} - \int_{\mathbb{R}^D} g_-(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}, \tag{B.2.49}$$

and both integrals are guaranteed to be non-negative since their integrands are.

In order to show that $h(\boldsymbol{x}_t)$ is well-defined, we need to show that $\int_{\mathbb{R}^D} g_+(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} < \infty$ or $\int_{\mathbb{R}^D} g_-(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} < \infty$. To show that $h(\boldsymbol{x}_t) > -\infty$, it merely suffices to show that $\int_{\mathbb{R}^D} g_-(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} < \infty$. To bound the integral of g_- we need to understand the quantity g_-, namely, we want to characterize when g is negative.

$$g(\boldsymbol{\xi}) \leq 0 \iff p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}) \geq 0 \iff \log p_t(\boldsymbol{\xi}) \geq 0 \iff p_t(\boldsymbol{\xi}) \geq 1. \tag{B.2.50}$$

Thus, it holds that

$$g_-(\boldsymbol{\xi}) = \mathbf{1}(p_t(\boldsymbol{\xi}) \geq 1) \cdot (-g(\boldsymbol{\xi})) = \mathbf{1}(p_t(\boldsymbol{\xi}) \geq 1) p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}). \tag{B.2.51}$$

In order to bound the integral of $g_-(\boldsymbol{\xi})$, we need to show that p_t is "not too concentrated," namely that p_t is not too large. To prove this, in this case we are lucky enough to be able to bound the function $g_-(\boldsymbol{\xi})$ itself. Namely, notice that

$$\max_{\boldsymbol{\xi} \in \mathbb{R}^D} \varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) = \varphi_t(\mathbf{0}) = \frac{1}{(2\pi)^{D/2} t^D} =: C_t. \tag{B.2.52}$$

which blows up as $t \to 0$ but is finite for all finite t. Therefore

$$p_t(\boldsymbol{\xi}) = \mathbb{E}\, \varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) \leq \mathbb{E}\, C_t = C_t. \tag{B.2.53}$$

Now there are two cases.

- If $C_t < 1$, then $p_t(\boldsymbol{\xi}) < 1$, so the indicator is never 1, hence $g_- = 0$ identically and its integral is also 0.
- If $C_t \geq 1$, then $\log C_t \geq 0$, so since the logarithm is monotonically increasing,

$$\begin{aligned} \int_{\mathbb{R}^D} g_-(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} &= \int_{\mathbb{R}^D} \mathbf{1}(p_t(\boldsymbol{\xi}) \geq 1) p_t(\boldsymbol{\xi}) \log p_t(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} && \text{(B.2.54)} \\ &= \mathbb{E}[\mathbf{1}(p_t(\boldsymbol{x}_t) \geq 1) \log p_t(\boldsymbol{x}_t)] && \text{(B.2.55)} \\ &\leq \mathbb{E}[\mathbf{1}(p_t(\boldsymbol{x}_t) \geq 1) \log C_t] && \text{(B.2.56)} \\ &= \mathbb{P}[p_t(\boldsymbol{x}_t) \geq 1] \log C_t. && \text{(B.2.57)} \end{aligned}$$

Hence we have $\int_{\mathbb{R}^D} g_-(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} < \infty$, so the differential entropy $h(\boldsymbol{x}_t)$ exists and is $> -\infty$.

To prove Lemma B.1.2, suppose that Assumption B.1 holds. We want to show that $h(\boldsymbol{x}) < \infty$ and $h(\boldsymbol{x}_t) < \infty$. The mechanism for doing this is the same, and involves the maximum entropy result Theorem B.1. Namely, since $\boldsymbol{x}$ is absolutely bounded, it has a finite covariance which we will denote $\boldsymbol{\Sigma}$. Then the covariance of $\boldsymbol{x}_t$ is $\boldsymbol{\Sigma} + t^2\boldsymbol{I}$. Thus the entropy of $\boldsymbol{x}$ and $\boldsymbol{x}_t$ can be upper bounded by the entropy of normal distributions with the respective covariances, i.e., $\log[(2\pi e)^D \det(\boldsymbol{\Sigma})]$ or $\log[(2\pi e)^D \det(\boldsymbol{\Sigma} + t^2\boldsymbol{I})]$, and both are $< \infty$. □

Integration by Parts in De Bruijn Identity

Finally, we fill in the integration-by-parts argument alluded to in the proofs of Theorems B.2 and B.3. The argument is conceptually pretty simple but requires some technical estimates to show that the boundary term in the integration-by-parts vanishes.

Lemma B.2. *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). For $t \geq 0$, let p_t be the density of $\boldsymbol{x}_t$. Then for a constant $c \in \mathbb{R}$ it holds*

$$\int_{\mathbb{R}^D} \Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})]\mathrm{d}\boldsymbol{\xi} = -\int_{\mathbb{R}^D} \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle \mathrm{d}\boldsymbol{\xi}. \tag{B.2.58}$$

Proof. The basic idea of this proof is in two steps:

- First, apply Green's theorem to do integration by parts over a compact set;
- Second, send the radius of this compact set to $+\infty$, to get integrals over all of $\mathbb{R}^D$.

Green's theorem says that for any compact set $\mathcal{K} \subseteq \mathbb{R}^D$, twice continuously differentiable $\phi\colon \mathbb{R}^D \to \mathbb{R}$, and continuously differentiable $\psi\colon \mathbb{R}^D \to \mathbb{R}$,

$$\int_{\mathcal{K}} \{\psi(\boldsymbol{\xi})\Delta\phi(\boldsymbol{\xi}) + \langle \nabla\psi(\boldsymbol{\xi}), \nabla\phi(\boldsymbol{\xi})\rangle\}\, \mathrm{d}\boldsymbol{\xi} = \int_{\partial\mathcal{K}} \psi(\boldsymbol{\xi})\langle \nabla\phi(\boldsymbol{\xi}), \boldsymbol{n}(\boldsymbol{\xi})\rangle \mathrm{d}\sigma(\boldsymbol{\xi}) \tag{B.2.59}$$

where $\mathrm{d}\sigma(\boldsymbol{\xi})$ denotes an integral over the "surface measure", i.e., the inherited measure on $\partial\mathcal{K}$, namely the boundary of $\mathcal{K}$, and accordingly $\boldsymbol{\xi}$ takes values on this surface and $\boldsymbol{n}(\boldsymbol{\xi})$ is the unit normal vector to $\mathcal{K}$ at the surface point $\boldsymbol{\xi}$. Now, taking $\phi(\boldsymbol{\xi}) \doteq p_t(\boldsymbol{\xi})$ and $\psi(\boldsymbol{\xi}) \doteq c + \log p_t(\boldsymbol{\xi})$, over a ball $B_r(\mathbf{0})$ of radius $r > 0$ centered at $\mathbf{0}$ (so that $\partial B_r(\mathbf{0})$ is the sphere of radius r centered at $\mathbf{0}$ and $\boldsymbol{n}(\boldsymbol{\xi}) = \boldsymbol{\xi}/\|\boldsymbol{\xi}\|_2 = \boldsymbol{\xi}/r$):

$$\int_{B_r(\mathbf{0})} \{\Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})] + \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle\}\, \mathrm{d}\boldsymbol{\xi} \tag{B.2.60}$$

$$= \int_{\partial B_r(\mathbf{0})} [c + \log p_t(\boldsymbol{\xi})] \left\langle \nabla p_t(\boldsymbol{\xi}), \frac{\boldsymbol{\xi}}{r} \right\rangle \mathrm{d}\sigma(\boldsymbol{\xi}) \tag{B.2.61}$$

$$= \frac{1}{r}\int_{\partial B_r(\mathbf{0})} [c + \log p_t(\boldsymbol{\xi})]\, \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle\, \mathrm{d}\sigma(\boldsymbol{\xi}). \tag{B.2.62}$$

Sending $r \to \infty$, it holds that

$$\int_{\mathbb{R}^D} \{\Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})] + \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle\} \, \mathrm{d}\boldsymbol{\xi} \tag{B.2.63}$$

$$= \lim_{r\to\infty} \int_{B_r(\mathbf{0})} \{\Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})] + \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle\} \, \mathrm{d}\boldsymbol{\xi} \tag{B.2.64}$$

$$= \lim_{r\to\infty} \frac{1}{r} \int_{\partial B_r(\mathbf{0})} [c + \log p_t(\boldsymbol{\xi})] \, \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \, \mathrm{d}\sigma(\boldsymbol{\xi}), \tag{B.2.65}$$

where the first equality follows by dominated convergence on the integrand. It remains to compute the last limit. For this, we take asymptotic expansions of each term. The main device is as follows: for $\boldsymbol{\xi} \in \partial B_r(\mathbf{0})$, we have $\|\boldsymbol{\xi}\|_2 = r$, so

$$p_t(\boldsymbol{\xi}) = \mathbb{E}[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x})] \tag{B.2.66}$$

$$= \mathbb{E}\left[\underbrace{\frac{1}{(2\pi)^{D/2} t^D}}_{\doteq C_t} e^{-\|\boldsymbol{\xi}-\boldsymbol{x}\|_2^2/(2t^2)} \right] \tag{B.2.67}$$

$$= C_t \, \mathbb{E}\left[e^{-(\|\boldsymbol{\xi}\|_2^2 - 2\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \|\boldsymbol{x}\|_2^2)/(2t^2)} \right] \tag{B.2.68}$$

$$= C_t \, \mathbb{E}\left[e^{-(r^2 - 2\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \|\boldsymbol{x}\|_2^2)/(2t^2)} \right] \tag{B.2.69}$$

$$= C_t e^{-r^2/(2t^2)} \, \mathbb{E}[e^{(2\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle - \|\boldsymbol{x}\|_2^2)/(2t^2)}]. \tag{B.2.70}$$

Note that because $\|\boldsymbol{\xi}\|_2 = r$, we have by Cauchy-Schwarz that

$$-2r\|\boldsymbol{x}\|_2 - \|\boldsymbol{x}\|_2^2 \le 2\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle - \|\boldsymbol{x}\|_2^2 \le 2r\|\boldsymbol{x}\|_2 - \|\boldsymbol{x}\|_2^2. \tag{B.2.71}$$

Recall that by Assumption B.1, $\boldsymbol{x}$ is supported on a compact set $\mathcal{S}$ of radius R. Thus

$$-2R(r + R) \le 2\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle - \|\boldsymbol{x}\|_2^2 \le 2Rr. \tag{B.2.72}$$

In other words, it holds

$$C_t e^{-[r^2 + 2R(r+R)]/(2t^2)} \le p_t(\boldsymbol{\xi}) \le C_t e^{[-r^2 + 2Rr]/(2t^2)}. \tag{B.2.73}$$

Now to compute the gradient, we can use Proposition B.1 and linearity of expectation to compute

$$\langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle = \left\langle -\frac{1}{t^2} \, \mathbb{E}[(\boldsymbol{\xi} - \boldsymbol{x}) \, \varphi_t(\boldsymbol{\xi} - \boldsymbol{x})], \boldsymbol{\xi} \right\rangle \tag{B.2.74}$$

$$= -\frac{1}{t^2} \, \mathbb{E}[\langle \boldsymbol{\xi} - \boldsymbol{x}, \boldsymbol{\xi}\rangle \varphi_t(\boldsymbol{\xi} - \boldsymbol{x})] \tag{B.2.75}$$

$$= -\frac{1}{t^2} \, \mathbb{E}\big[\big(\|\boldsymbol{\xi}\|_2^2 - \langle \boldsymbol{\xi}, \boldsymbol{x}\rangle\big) \, \varphi_t(\boldsymbol{\xi} - \boldsymbol{x})\big] \tag{B.2.76}$$

$$= -\frac{1}{t^2} \, \mathbb{E}\big[\big(r^2 - \langle \boldsymbol{\xi}, \boldsymbol{x}\rangle\big) \, \varphi_t(\boldsymbol{\xi} - \boldsymbol{x})\big] \tag{B.2.77}$$

$$= \frac{1}{t^2}\,\mathbb{E}\big[\big(\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle - r^2\big)\,\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\big]. \tag{B.2.78}$$

Using Cauchy-Schwarz and the representation $p_t(\boldsymbol{\xi}) \doteq \mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})]$ again, it holds

$$\frac{1}{t^2}\,\mathbb{E}\big[\big(-Rr - r^2\big)\,\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\big] \le \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \le \frac{1}{t^2}\,\mathbb{E}\big[\big(Rr - r^2\big)\,\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\big] \tag{B.2.79}$$

$$\frac{1}{t^2}\,\big(-Rr - r^2\big)\,\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})] \le \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \le \frac{1}{t^2}\,\big(Rr - r^2\big)\,\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})] \tag{B.2.80}$$

$$-\frac{r(R+r)}{t^2} p_t(\boldsymbol{\xi}) \le \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \le -\frac{r(r-R)}{t^2} p_t(\boldsymbol{\xi}). \tag{B.2.81}$$

For $r > R > 0$ (as is suitable, because we are going to take the limit $r \to \infty$ while R is fixed), both sides are negative. This makes sense: most of the probability mass is contained within the ball of radius R and thus the score points inwards, having a negative inner product with the outward-pointing vector $\boldsymbol{\xi}$. Thus using the appropriate bounds for $p_t(\boldsymbol{\xi})$,

$$-\frac{r(R+r)}{t^2}\cdot C_t e^{[-r^2+2Rr]/(2t^2)} \le \langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \le -\frac{r(r-R)}{t^2}\cdot C_t e^{-[r^2+2R(r+R)]/(2t^2)}. \tag{B.2.82}$$

Then, noting that $C_t = \mathrm{poly}(t^{-1})$, we can compute

$$[c + \log p_t(\boldsymbol{\xi})]\langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle = \mathrm{poly}(r, R, t^{-1}, c) e^{-\Theta_r(r^2)} \tag{B.2.83}$$

So one can see that, letting the surface area of $\partial B_r(\mathbf{0})$ be $\omega_D r^{D-1}$ where ω_D is a function of D, it holds

$$\frac{1}{r}\int_{\partial B_r(\mathbf{0})} [c + \log p_t(\boldsymbol{\xi})]\langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \mathrm{d}\boldsymbol{\xi} = \mathrm{poly}(r, R, t^{-1}, c) e^{-\Theta_r(r^2)} \tag{B.2.84}$$

and therefore the exponentially decaying tails mean

$$\lim_{r\to\infty} \frac{1}{r}\int_{\partial B_r(\mathbf{0})} [c + \log p_t(\boldsymbol{\xi})]\langle \nabla p_t(\boldsymbol{\xi}), \boldsymbol{\xi}\rangle \mathrm{d}\boldsymbol{\xi} = 0. \tag{B.2.85}$$

Finally, plugging into (B.2.63), we have

$$\int_{\mathbb{R}^D} \{\Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})] + \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle\}\,\mathrm{d}\boldsymbol{\xi} = 0 \tag{B.2.86}$$

$$\implies \int_{\mathbb{R}^D} \Delta p_t(\boldsymbol{\xi})[c + \log p_t(\boldsymbol{\xi})]\mathrm{d}\boldsymbol{\xi} = -\int_{\mathbb{R}^D} \langle \nabla \log p_t(\boldsymbol{\xi}), \nabla p_t(\boldsymbol{\xi})\rangle \mathrm{d}\boldsymbol{\xi} \tag{B.2.87}$$

as claimed. □

Local Invertibility of the Denoiser $\bar{\boldsymbol{x}}$

Here we provide some results used in the proof of Theorem B.3 which are appropriate generalizations of corresponding results in [Gri11].

Lemma B.3 (Generalization of [Gri11], Lemma A.1)**.** *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). Let $s,t\in[0,T]$ be such that $0\le s<t\le T$, and let $\bar{\boldsymbol{x}}(\boldsymbol{\xi})\doteq\mathbb{E}[\boldsymbol{x}_s\mid\boldsymbol{x}_t=\boldsymbol{\xi}]$. The Jacobian $\bar{\boldsymbol{x}}'(\boldsymbol{\xi})$ is symmetric positive definite.*

Proof. We have

$$\bar{\boldsymbol{x}}'(\boldsymbol{\xi})=\boldsymbol{I}+\left(1-\frac{s}{t}\right)t^2\nabla^2\log p_t(\boldsymbol{\xi}). \tag{B.2.88}$$

Here we expand

$$\nabla^2\log p_t(\boldsymbol{\xi})=\frac{p_t(\boldsymbol{\xi})\nabla^2p_t(\boldsymbol{\xi})-(\nabla p_t(\boldsymbol{\xi}))(\nabla p_t(\boldsymbol{\xi}))^\top}{p_t(\boldsymbol{\xi})^2}. \tag{B.2.89}$$

So we need to ensure that

$$\bar{\boldsymbol{x}}'(\boldsymbol{\xi})=\boldsymbol{I}+\left(1-\frac{s}{t}\right)t^2\frac{p_t(\boldsymbol{\xi})\nabla^2p_t(\boldsymbol{\xi})-(\nabla p_t(\boldsymbol{\xi}))(\nabla p_t(\boldsymbol{\xi}))^\top}{p_t(\boldsymbol{\xi})^2} \tag{B.2.90}$$

$$=\frac{p_t(\boldsymbol{\xi})^2\boldsymbol{I}+\left(1-\frac{s}{t}\right)t^2\left[p_t(\boldsymbol{\xi})\nabla^2p_t(\boldsymbol{\xi})-(\nabla p_t(\boldsymbol{\xi}))(\nabla p_t(\boldsymbol{\xi}))^\top\right]}{p_t(\boldsymbol{\xi})^2} \tag{B.2.91}$$

is symmetric positive semidefinite. Indeed it is obviously symmetric (by Clairaut's theorem). To show its positive semidefiniteness, we plug in the expectation representation of p_t given by (B.2.3) (and ∇p_t, Δp_t by Proposition B.1) to obtain (where $\boldsymbol{x}$ is as defined and $\boldsymbol{y}$ is i.i.d. as $\boldsymbol{x}$),

$$\boldsymbol{v}^\top[\bar{\boldsymbol{x}}'(\boldsymbol{\xi})]\boldsymbol{v} \tag{B.2.92}$$

$$=p_t(\boldsymbol{\xi})^{-2}\boldsymbol{v}^\top\Bigg\{p_t(\boldsymbol{\xi})^2\boldsymbol{I}+\left(1-\frac{s}{t}\right)t^2\,\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})]\,\mathbb{E}\left[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\cdot\frac{(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{x})^\top-t^2\boldsymbol{I}}{t^4}\right] \tag{B.2.93}$$

$$-\left(1-\frac{s}{t}\right)t^2\,\mathbb{E}\left[-\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\cdot\frac{\boldsymbol{\xi}-\boldsymbol{x}}{t^2}\right]\mathbb{E}\left[-\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\cdot\frac{\boldsymbol{\xi}-\boldsymbol{x}}{t^2}\right]^\top\Bigg\}\boldsymbol{v} \tag{B.2.94}$$

$$=p_t(\boldsymbol{\xi})^{-2}\boldsymbol{v}^\top\Bigg\{\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\varphi_t(\boldsymbol{\xi}-\boldsymbol{y})\boldsymbol{I}] \tag{B.2.95}$$

$$+\left(1-\frac{s}{t}\right)t^2\,\mathbb{E}\left[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\varphi_t(\boldsymbol{\xi}-\boldsymbol{y})\cdot\frac{(\boldsymbol{\xi}-\boldsymbol{y})(\boldsymbol{\xi}-\boldsymbol{y})^\top-t^2\boldsymbol{I}}{t^4}\right] \tag{B.2.96}$$

$$-\left(1-\frac{s}{t}\right)t^2\,\mathbb{E}\left[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\varphi_t(\boldsymbol{\xi}-\boldsymbol{y})\cdot\frac{(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{y})^\top}{t^4}\right]\Bigg\} \tag{B.2.97}$$

$$= \frac{1-s/t}{p_t(\boldsymbol{\xi})^2} \boldsymbol{v}^\top \mathbb{E}\left[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\varphi_t(\boldsymbol{\xi}-\boldsymbol{y})\left\{\frac{1}{1-s/t}\boldsymbol{I} + \frac{(\boldsymbol{\xi}-\boldsymbol{y})(\boldsymbol{\xi}-\boldsymbol{y})^\top}{t^2} - \boldsymbol{I} - \frac{(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{y})^\top}{t^2}\right\}\right]\boldsymbol{v} \tag{B.2.98}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \boldsymbol{v}^\top \mathbb{E}\left[\frac{s}{t-s}\boldsymbol{I} + \frac{(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{x})^\top}{2t^2} + \frac{(\boldsymbol{\xi}-\boldsymbol{y})(\boldsymbol{\xi}-\boldsymbol{y})^\top}{2t^2} - \frac{(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{y})^\top}{t^2}\right]\boldsymbol{v} \tag{B.2.99}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \boldsymbol{v}^\top \mathbb{E}\left[\frac{s}{t-s}\boldsymbol{I} + \frac{1}{2t^2}\left((\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{x})^\top + (\boldsymbol{\xi}-\boldsymbol{y})(\boldsymbol{\xi}-\boldsymbol{y})^\top - 2(\boldsymbol{\xi}-\boldsymbol{x})(\boldsymbol{\xi}-\boldsymbol{y})^\top\right)\right]\boldsymbol{v} \tag{B.2.100}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \mathbb{E}\left[\frac{s}{t-s}\|\boldsymbol{v}\|_2^2 + \frac{1}{2t^2}\left([(\boldsymbol{\xi}-\boldsymbol{x})^\top\boldsymbol{v}]^2 + [(\boldsymbol{\xi}-\boldsymbol{y})^\top\boldsymbol{v}]^2 - 2[(\boldsymbol{\xi}-\boldsymbol{x})^\top\boldsymbol{v}][(\boldsymbol{\xi}-\boldsymbol{y})^\top\boldsymbol{v}]\right)\right] \tag{B.2.101}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \mathbb{E}\left[\frac{s}{t-s}\|\boldsymbol{v}\|_2^2 + \frac{1}{2t^2}\left([(\boldsymbol{\xi}-\boldsymbol{x})^\top\boldsymbol{v}]^2 + [(\boldsymbol{\xi}-\boldsymbol{y})^\top\boldsymbol{v}]^2 - 2[(\boldsymbol{\xi}-\boldsymbol{x})^\top\boldsymbol{v}][(\boldsymbol{\xi}-\boldsymbol{y})^\top\boldsymbol{v}]\right)\right] \tag{B.2.102}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \mathbb{E}\left[\frac{s}{t-s}\|\boldsymbol{v}\|_2^2 + \frac{1}{2t^2}\left([(\boldsymbol{\xi}-\boldsymbol{x})^\top\boldsymbol{v}] - [(\boldsymbol{\xi}-\boldsymbol{y})^\top\boldsymbol{v}]\right)^2\right] \tag{B.2.103}$$

$$= \frac{t-s}{tp_t(\boldsymbol{\xi})^2} \mathbb{E}\left[\frac{s}{t-s}\|\boldsymbol{v}\|_2^2 + \frac{1}{2t^2}[(\boldsymbol{y}-\boldsymbol{x})^\top\boldsymbol{v}]^2\right] \tag{B.2.104}$$

$$= \frac{s}{tp_t(\boldsymbol{\xi})^2}\|\boldsymbol{v}\|_2^2 + \frac{t-s}{2t^3 p_t(\boldsymbol{\xi})} \mathbb{E}[\{(\boldsymbol{y}-\boldsymbol{x})^\top\boldsymbol{v}\}^2] \tag{B.2.105}$$

Since $\boldsymbol{x}$ and $\boldsymbol{y}$ are i.i.d., the whole integral (i.e., the original quadratic form) is 0 if and only if $s = 0$ and $\boldsymbol{x}$ has support entirely contained in an affine subspace which is orthogonal to $\boldsymbol{v}$. But this is ruled out by assumption (i.e., that $\boldsymbol{x}$ has a density on $\mathbb{R}^D$), so the Jacobian $\bar{\boldsymbol{x}}'(\boldsymbol{\xi})$ is symmetric positive definite. □

Lemma B.4 (Generalization of [Gri11] Corollary A.2, Part 1)**.** *Let $f\colon \mathbb{R}^D \to \mathbb{R}^D$ be any differentiable function whose Jacobian $f'(\boldsymbol{x})$ is symmetric positive definite. Then f is injective, and hence invertible as a function $\mathbb{R}^D \to \mathcal{R}(f)$ where $\mathcal{R}(f)$ is the range of f.*

Proof. Suppose that f were not injective, i.e., there exists $\boldsymbol{x}, \boldsymbol{x}'$ such that $f(\boldsymbol{x}) = f(\boldsymbol{x}')$ while $\boldsymbol{x} \neq \boldsymbol{x}'$. Define $\boldsymbol{v} \doteq (\boldsymbol{x}'-\boldsymbol{x})/\|\boldsymbol{x}'-\boldsymbol{x}\|_2$. Define the function $g\colon \mathbb{R} \to \mathbb{R}$ as $g(t) \doteq \boldsymbol{v}^\top f(\boldsymbol{x}+t\boldsymbol{v})$. Then $g(0) = \boldsymbol{v}^\top f(\boldsymbol{x}) = \boldsymbol{v}^\top f(\boldsymbol{x}') = g(\|\boldsymbol{x}'-\boldsymbol{x}\|_2)$. Since f is differentiable, g is differentiable, so the derivative g' must vanish for some $t^* \in (0, \|\boldsymbol{x}'-\boldsymbol{x}\|_2)$ by the mean value theorem. However,

$$g'(t^*) \doteq \boldsymbol{v}^\top \left[f'(\boldsymbol{x}+t^*\boldsymbol{v})\right]\boldsymbol{v} > 0 \tag{B.2.106}$$

since the Jacobian is positive definite. Thus we arrive at a contradiction, as claimed. □

Combining the above two results, we obtain the following crucial result.

Corollary B.1 (Generalization of [Gri11] Corollary A.2, Part 2)**.** *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). Let $s,t\in[0,T]$ be such that $0\leq s<t\leq T$, and let $\bar{\boldsymbol{x}}(\boldsymbol{\xi})\doteq\mathbb{E}[\boldsymbol{x}_s\mid\boldsymbol{x}_t=\boldsymbol{\xi}]$. Then $\bar{\boldsymbol{x}}$ is injective, and therefore invertible onto its range.*

Proof. The only thing left to show is that $\bar{\boldsymbol{x}}$ is differentiable, but this is immediate from Tweedie's formula (Theorem 3.2) which shows that $\bar{\boldsymbol{x}}$ is differentiable if and only if $\nabla\log p_t$ is differentiable, and this is provided by Equation (B.2.3). □

Controlling the Laplacian $\Delta\log p_t$

Finally, we develop a technical estimate which is required for the proof of Theorem B.3 and actually motivates the assumption for the viable t.

Lemma B.5. *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). Let p_t be the density of $\boldsymbol{x}_t$. Then, for $t>0$ it holds*

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^D}|\Delta\log p_t(\boldsymbol{\xi})|\leq\max\left(\frac{D}{t^2},\left|\frac{R^2}{t^4}-\frac{D}{t^2}\right|\right). \tag{B.2.107}$$

Proof. By chain rule, a simple exercise computes

$$\Delta\log p_t(\boldsymbol{\xi})=\frac{\Delta p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})}-\frac{\|\nabla p_t(\boldsymbol{\xi})\|_2^2}{p_t(\boldsymbol{\xi})^2}. \tag{B.2.108}$$

Using Proposition B.1 to write the terms in $\Delta p_t(\boldsymbol{\xi})$, we obtain

$$\frac{\Delta p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})}=\frac{\mathbb{E}\left[\frac{\|\boldsymbol{\xi}-\boldsymbol{x}\|_2^2-Dt^2}{t^4}\cdot\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\right]}{\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})]} \tag{B.2.109}$$

$$=\frac{\int_{\mathbb{R}^D}\left\{\frac{\|\boldsymbol{\xi}-\boldsymbol{u}\|_2^2-Dt^2}{t^4}\right\}\varphi_t(\boldsymbol{\xi}-\boldsymbol{u})p(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}{\int_{\mathbb{R}^D}\varphi_t(\boldsymbol{\xi}-\boldsymbol{u})p(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}. \tag{B.2.110}$$

This looks like a Bayesian marginalization, so let us define the appropriate normalized density

$$q_{\boldsymbol{\xi}}(\boldsymbol{u})=\frac{\varphi_t(\boldsymbol{\xi}-\boldsymbol{u})p(\boldsymbol{u})}{\int_{\mathbb{R}^D}\varphi_t(\boldsymbol{\xi}-\boldsymbol{v})p(\boldsymbol{v})\mathrm{d}\boldsymbol{v}}=\frac{\varphi_t(\boldsymbol{\xi}-\boldsymbol{u})p(\boldsymbol{u})}{\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})]}=\frac{\varphi_t(\boldsymbol{\xi}-\boldsymbol{u})p(\boldsymbol{u})}{p_t(\boldsymbol{\xi})} \tag{B.2.111}$$

Then, defining $\boldsymbol{y}_{\boldsymbol{\xi}}\sim q_{\boldsymbol{\xi}}$, we can write

$$\frac{\Delta p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})}=\int_{\mathbb{R}^D}\left\{\frac{\|\boldsymbol{\xi}-\boldsymbol{u}\|_2^2-Dt^2}{t^4}\right\}q_{\boldsymbol{\xi}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}=\frac{1}{t^4}\,\mathbb{E}[\|\boldsymbol{\xi}-\boldsymbol{y}_{\boldsymbol{\xi}}\|_2^2]-\frac{D}{t^2}. \tag{B.2.112}$$

Similarly, writing out the second term (non-squared) we obtain

$$\frac{\nabla p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})}=-\frac{\boldsymbol{\xi}-\mathbb{E}[\boldsymbol{y}_{\boldsymbol{\xi}}]}{t^2}. \tag{B.2.113}$$

Letting $\boldsymbol{z}_{\boldsymbol{\xi}} \doteq \boldsymbol{y}_{\boldsymbol{\xi}} - \boldsymbol{\xi}$, it holds

$$\frac{\Delta p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})} = \frac{\mathbb{E}[\|\boldsymbol{z}_{\boldsymbol{\xi}}\|_2^2]}{t^4} - \frac{D}{t^2}, \qquad \frac{\nabla p_t(\boldsymbol{\xi})}{p_t(\boldsymbol{\xi})} = \frac{\mathbb{E}[\boldsymbol{z}_{\boldsymbol{\xi}}]}{t^2}. \tag{B.2.114}$$

Thus writing $\Delta \log p_t$ out fully, we have

$$\Delta \log p_t(\boldsymbol{\xi}) = \frac{\mathbb{E}[\|\boldsymbol{z}_{\boldsymbol{\xi}}\|_2^2]}{t^4} - \frac{D}{t^2} - \frac{\|\,\mathbb{E}[\boldsymbol{z}_{\boldsymbol{\xi}}]\|_2^2}{t^4} \tag{B.2.115}$$

$$= \frac{\mathbb{E}[\|\boldsymbol{z}_{\boldsymbol{\xi}}\|_2^2] - \|\,\mathbb{E}[\boldsymbol{z}_{\boldsymbol{\xi}}]\|_2^2}{t^4} - \frac{D}{t^2} \tag{B.2.116}$$

$$= \frac{\operatorname{tr}(\operatorname{Cov}(\boldsymbol{z}_{\boldsymbol{\xi}}))}{t^4} - \frac{D}{t^2} \tag{B.2.117}$$

$$= \frac{\operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}_{\boldsymbol{\xi}}))}{t^4} - \frac{D}{t^2}. \tag{B.2.118}$$

A trivial lower bound on this trace is 0, since covariance matrices are positive semidefinite. To find an upper bound, note that $\boldsymbol{y}_{\boldsymbol{\xi}}$ takes values only in the support of $\boldsymbol{x}$ (since p is a factor of the density $q_{\boldsymbol{\xi}}$ of $\boldsymbol{y}_{\boldsymbol{\xi}}$), which by Assumption B.1 is a compact set $\mathcal{S}$ with radius $R \doteq \sup_{\boldsymbol{\xi}\in\mathcal{S}} \|\boldsymbol{\xi}\|_2$. So

$$\operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}_{\boldsymbol{\xi}})) = \mathbb{E}[\|\boldsymbol{y}_{\boldsymbol{\xi}}\|_2^2] - \|\,\mathbb{E}[\boldsymbol{y}_{\boldsymbol{\xi}}]\|_2^2 \leq \mathbb{E}[\|\boldsymbol{y}_{\boldsymbol{\xi}}\|_2^2] \leq R^2. \tag{B.2.119}$$

Therefore

$$-\frac{D}{t^2} \leq \Delta \log p_t(\boldsymbol{\xi}) \leq \frac{R^2}{t^4} - \frac{D}{t^2}, \tag{B.2.120}$$

which shows the claim. □

Derivative Computations

Here we calculate some useful derivatives which will be reused throughout the appendix.

Proposition B.1. *Let $\boldsymbol{x}$ be any random variable such that Assumptions B.1 and B.2 hold, and let $(\boldsymbol{x}_t)_{t\in[0,T]}$ be the stochastic process (B.2.1). For $t \geq 0$, let p_t be the density of $\boldsymbol{x}_t$. Then*

$$\frac{\partial p_t}{\partial t}(\boldsymbol{\xi}) = \mathbb{E}\left[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) \cdot \frac{\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 - Dt^2}{t^3}\right] \tag{B.2.121}$$

$$\nabla p_t(\boldsymbol{\xi}) = -\,\mathbb{E}\left[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) \cdot \frac{\boldsymbol{\xi} - \boldsymbol{x}}{t^2}\right] \tag{B.2.122}$$

$$\nabla^2 p_t(\boldsymbol{\xi}) = \mathbb{E}\left[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) \cdot \frac{(\boldsymbol{\xi} - \boldsymbol{x})(\boldsymbol{\xi} - \boldsymbol{x})^\top - t^2\boldsymbol{I}}{t^4}\right] \tag{B.2.123}$$

$$\Delta p_t(\boldsymbol{\xi}) = \mathbb{E}\left[\varphi_t(\boldsymbol{\xi} - \boldsymbol{x}) \cdot \frac{\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 - Dt^2}{t^4}\right]. \tag{B.2.124}$$

Proof. We use the convolution representation of p_t, namely (B.2.3). First taking the time derivative, a computation yields that Proposition B.3 applies,[5] so we can bring the derivative inside the integral/expectation as:

$$\frac{\partial p_t}{\partial t}(\boldsymbol{\xi}) = \frac{\partial}{\partial t}\,\mathbb{E}[\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})] = \mathbb{E}\left[\frac{\partial}{\partial t}\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})\right] = \frac{\partial \varphi_t}{\partial t} * p. \tag{B.2.125}$$

Meanwhile, by properties of convolutions (Proposition B.4) and using the fact that p is compactly supported (Assumption B.1),

$$p_t = \varphi_t * p \implies \nabla p_t = \nabla\varphi_t * p \implies \nabla^2 p_t = \nabla^2\varphi_t * p \implies \Delta p_t = \Delta\varphi_t * p. \tag{B.2.126}$$

The rest of the computation follows from Proposition B.2. □

Proposition B.2. *For $t > 0$ and $\boldsymbol{\xi} \in \mathbb{R}^D$ it holds*

$$\frac{\partial}{\partial t}\varphi_t(\boldsymbol{\xi}) = \varphi_t(\boldsymbol{\xi}) \cdot \frac{\|\boldsymbol{\xi}\|_2^2 - Dt^2}{t^3} \tag{B.2.127}$$

$$\nabla\varphi_t(\boldsymbol{\xi}) = -\varphi_t(\boldsymbol{\xi}) \cdot \frac{\boldsymbol{\xi}}{t^2} \tag{B.2.128}$$

$$\nabla^2\varphi_t(\boldsymbol{\xi}) = \varphi_t(\boldsymbol{\xi}) \cdot \frac{\boldsymbol{\xi}\boldsymbol{\xi}^\top - t^2\boldsymbol{I}}{t^4} \tag{B.2.129}$$

$$\Delta\varphi_t(\boldsymbol{\xi}) = \varphi_t(\boldsymbol{\xi}) \cdot \frac{\|\boldsymbol{\xi}\|_2^2 - Dt^2}{t^4}. \tag{B.2.130}$$

Proof. Direct computation. □

Differentiating Under the Integral Sign

In this appendix, we differentiate under the integral sign many times, and it is important to know when we can do this. There are two kinds of differentiating under the integral sign:

1. Differentiating an integral $\int f_t(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi}$ with respect to the auxiliary parameter t.
2. Differentiating a convolution $(f * g)(\boldsymbol{\xi}) = \int f(\boldsymbol{\xi})g(\boldsymbol{\xi}-\boldsymbol{u})\mathrm{d}\boldsymbol{u}$ with respect to the variable $\boldsymbol{\xi}$.

For the first category, we give a concrete result, stated without proof but attributable to the linked source, which derives the following result as a special case of a more general theorem about the interaction of differential operators and tempered distributions, much beyond the scope of the book. A full formal reference can be found in [Jon82].

[5] We use $f_t(\boldsymbol{\xi}) = p(\boldsymbol{\xi})\varphi_t(\boldsymbol{\xi}-\boldsymbol{x})$, noting that it is twice continuously differentiable in $\boldsymbol{\xi}$ and (more than) twice continuously differentiable in t. Then to check the local integrability of f_t we compute $\frac{\partial f_t}{\partial t}(\boldsymbol{\xi}) = f_t(\boldsymbol{\xi}) \cdot \frac{1}{t^3}(\|\boldsymbol{\xi}-\boldsymbol{x}\|_2^2 - Dt^2)$, which is is easy to check integrable over $\boldsymbol{\xi}$ and $t \in [t_{\min}, t_{\max}]$ where $t_{\min} > 0$. (Indeed, f_t has exponentially decaying tails, so the quadratic term in the product is of no issue.)

Proposition B.3 ([Jon82], Section 11.12)**.** *Let $f\colon (0,T)\times\mathbb{R}^D\to\mathbb{R}$ be such that:*

- *f is a jointly measurable function of $(t,\boldsymbol{\xi})$;*
- *For Lebesgue-almost every $\boldsymbol{\xi}\in\mathbb{R}^D$, the function $t\mapsto f_t(\boldsymbol{\xi})$ is absolutely continuous;*
- *$\frac{\partial f_t}{\partial t}$ is locally integrable, i.e., for every $[t_{\min},t_{\max}]\subseteq(0,T)$ it holds*

$$\int_{t_{\min}}^{t_{\max}}\int_{\mathbb{R}^D}\left|\frac{\partial f_t}{\partial t}(\boldsymbol{\xi})\right|\mathrm{d}\boldsymbol{\xi}<\infty. \tag{B.2.131}$$

Then $t\mapsto\int_{\mathbb{R}^D}f_t(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi}$ is an absolutely continuous function on $(0,T)$, and its derivative is

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^D}f_t(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi}=\int_{\mathbb{R}^D}\frac{\partial}{\partial t}f_t(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi}, \tag{B.2.132}$$

defined for almost every $t\in(0,T)$.

For the second category, we give another concrete result, stated without proof but fully formalized in [BB11].

Proposition B.4 ([BB11], Proposition 4.20)**.** *Let f be k-times continuously differentiable with compact support, and let g be locally integrable. Then the convolution $f*g$ defined by*

$$(f*g)(\boldsymbol{\xi})\doteq\int_{\mathbb{R}^D}f(\boldsymbol{u})g(\boldsymbol{\xi}-\boldsymbol{u})\mathrm{d}\boldsymbol{u} \tag{B.2.133}$$

is k-times continuously differentiable, and its derivative of order k is

$$\nabla^k(f*g)=(\nabla^k f)*g. \tag{B.2.134}$$

Although not in the book, a simple integration by parts argument shows that if g is also k-times differentiable, then we can "trade off" the regularity:

$$\nabla^k(f*g)=f*(\nabla^k g). \tag{B.2.135}$$

B.3 Lossy Coding and Sphere Packing

In this section, we prove Theorem 4.1. Following our conventions throughout this appendix, we write $\mathcal{S}=\mathrm{Supp}(\boldsymbol{x})$ for the compact support of the random variable $\boldsymbol{x}$.

As foreshadowed, we will make a regularity assumption on the support set $\mathcal{S}$ to prove Theorem 4.1. One possibility for proceeding under minimal assumptions would be to instantiate the results of [RBK18; RKB23] in our setting, since these results apply to sets $\mathcal{S}$ with very low regularity (e.g., Cantor-like sets with fractal structure). However, we have found precisely computing the constants

in these results, a necessary endeavor to assert a conclusion like Theorem 4.1, to be somewhat onerous in our setting. Our approach is therefore to add a geometric regularity assumption on the set $\mathcal{S}$ that sacrifices some generality, but allows us to develop a more transparent argument. To avoid sacrificing too much generality, we must ensure that low-dimensionality in the set $\mathcal{S}$ is not prohibited. We therefore consider the running example we have used throughout the book, the mixture of low-rank Gaussian distributions. In this geometric setting, we model $\mathcal{S}$ as a union of hyperspheres, each of dimension d_k (possibly much smaller than D), living in mutually orthogonal subspaces of $\mathbb{R}^D$. This is a geometric simplification of the Gaussian mixture model: each component's support is approximated by a sphere in the subspace spanned by its principal directions. When the component dimensions d_k are large, this assumption is equivalent to a mixture of low-rank Gaussians assumption, by high-dimensional measure concentration. The CRATE models we develop and train in Chapters 5 and 8 have their *subspace dimensions* set compatibly with this assumption.

Assumption B.3. The support $\mathcal{S} \subset \mathbb{R}^D$ of the random variable $\boldsymbol{x}$ is a finite union of K spheres, each with dimension d_k, $k \in [K]$. The probability that $\boldsymbol{x}$ is drawn from the k-th sphere is given by $\pi_k \in [0, 1]$, and conditional on being drawn from the k-th sphere, $\boldsymbol{x}$ is uniformly distributed on that sphere. The supports satisfy that each sphere is mutually orthogonal with all others.

We proceed under the simplifying Assumption B.3 in order to simplify excessive technicality, and to connect to an important running example used throughout the monograph. We believe our results can be generalized to support $\mathcal{S}$ from the class of *sets with positive reach* with additional technical effort, but leave this for the future.

B.3.1 Proof of Relationship Between Rate Distortion and Covering

We briefly sketch the proof, then proceed to establishing three fundamental lemmas, then give the proof. The proof will depend on notions introduced in the sketch below.

Obtaining an upper bound on the rate distortion function (4.1.9) is straightforward: by the rate characterization (i.e., the rate distortion function is the minimum rate of a code for $\boldsymbol{x}$ with expected squared ℓ^2 distortion ϵ), upper bounding $\mathcal{R}_\epsilon(\boldsymbol{x})$ only requires demonstrating one code for $\boldsymbol{x}$ that achieves this target distortion, and any ϵ-covering of $\mathrm{Supp}(\boldsymbol{x})$ achieves this, with rate equal to the base-2 logarithm of the cardinality of the covering. The lower bound is more subtle. We make use of the Shannon lower bound, discussed in Remark 4.3: working out the constants in [LZ94, §III, (22)] gives a more precise version of the result quoted in Equation (4.1.17) (in bits, of course): for any random variable $\boldsymbol{x}$ with compact support and a density, it holds

$$\mathcal{R}_\epsilon(\boldsymbol{x}) \geq h(\boldsymbol{x}) - \log \mathrm{vol}(B_\epsilon) + \log \left(\frac{2}{D\Gamma(D/2)} \left(\frac{D}{2e} \right)^{D/2} \right), \tag{B.3.1}$$

with entropy (etc.) in nats in this expression. The constant can be easily estimated using Stirling's approximation. A quantitative form of Stirling's approximation which is often useful gives for any $x > 0$ [Jam15]

$$\Gamma(x) \leq \sqrt{2\pi} x^{x-1/2} e^{-x} e^{1/(12x)}. \tag{B.3.2}$$

We will apply this bound to $\Gamma(D/2)$ in Equation (B.3.1). We get

$$\log\left(\frac{2}{D\Gamma(D/2)}\left(\frac{D}{2e}\right)^{D/2}\right) \geq -\frac{1}{6D} + \log\left(\frac{2}{D}\left(\frac{D}{2e}\right)^{D/2} \cdot \sqrt{\frac{D}{4\pi}}\left(\frac{D}{2e}\right)^{-D/2}\right) \tag{B.3.3}$$

$$= -\frac{1}{6D} - \frac{1}{2}\log D\pi, \tag{B.3.4}$$

which we can take for the explicit value of the constant C_D in Equation (4.1.17). Summarizing the fully quantified Shannon lower bound (in bits):

$$\mathcal{R}_\epsilon(\boldsymbol{x}) \geq h(\boldsymbol{x}) - \log_2 \operatorname{vol}(B_\epsilon) - O(\log D). \tag{B.3.5}$$

Now, the important constraint for our current purposes is that the Shannon lower bound requires the random variable $\boldsymbol{x}$ to have a density, which rules out many low-dimensional distributions of interest. But let us momentarily consider the situation when $\boldsymbol{x}$ does admit a density. The assumption that $\boldsymbol{x}$ is uniformly distributed on its support is easily formalized in this setting: for any Borel set $A \subset \mathcal{S}$, we have

$$\mathbb{P}[\boldsymbol{x} \in A] = \int_A \frac{1}{\operatorname{vol}(\mathcal{S})} \mathrm{d}\boldsymbol{x}. \tag{B.3.6}$$

Then the entropy $h(\boldsymbol{x})$ is just

$$h(\boldsymbol{x}) = \log_2 \operatorname{vol}(\mathcal{S}). \tag{B.3.7}$$

The proof then concludes with a lemma that relates the ratio $\operatorname{vol}(\mathcal{S})/\operatorname{vol}(B_\epsilon)$ to the ϵ-covering number of $\mathcal{S}$ by ϵ balls.

To extend the program above to degenerate distributions satisfying Assumption B.3, our proof of the lower bound in Theorem 4.1 will leverage an approximation argument of the actual low-dimensional distribution $\boldsymbol{x}$ by "nearby" distributions which have densities, similarly but not exactly the same as the proof sketch preceding Theorem B.1. We will then link the parameter introduced in the approximating sequence to the distortion parameter ϵ in order to obtain the desired conclusion in Theorem 4.1.

Definition B.1. Let $\mathcal{S}$ be a compact set. For any $\delta > 0$, define the δ-thickening of $\mathcal{S}$, denoted $\mathcal{S}_\delta$, by

$$\mathcal{S}_\delta = \left\{\boldsymbol{\xi} \in \mathbb{R}^D \;\middle|\; \operatorname{dist}(\boldsymbol{\xi}, \mathcal{S}) \leq \delta\right\}. \tag{B.3.8}$$

The distance function referenced in Definition B.1 is defined by

$$\operatorname{dist}(\boldsymbol{\xi}, \mathcal{S}) = \inf_{\boldsymbol{\xi}' \in \mathcal{S}} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_2 . \tag{B.3.9}$$

For a compact set $\mathcal{S}$, Weierstrass's theorem implies that for any $\boldsymbol{\xi} \in \mathbb{R}^D$, there is always some $\boldsymbol{\xi}' \in \mathcal{S}$ attaining the infimum in the distance function. Compactness of $\mathcal{S}_\delta$ follows readily from compactness of $\mathcal{S}$, so $\operatorname{vol}(\mathcal{S}_\delta)$ is finite for any $\delta > 0$. It is then possible to make the following definition of a thickened random variable, specialized to Assumption B.3.

Definition B.2. Let $\boldsymbol{x}$ be a random variable such that $\operatorname{Supp}(\boldsymbol{x}) = \mathcal{S}$ is a union of K hyperspheres, distributed as in Assumption B.3. Denote the support of each component of the mixture by $\mathcal{S}_k$. Define the thickened random variable $\boldsymbol{x}_\delta$ as the mixture of measures where each component measure is uniform on the thickened set $\mathcal{S}_{k,\delta}$ (Definition B.1), for $k \in [K]$, with mixing weights π_k.

Lemma B.6. *Suppose the random variable $\boldsymbol{x}$ satisfies Assumption B.3. Then if $0 < \delta < \frac{1}{2}$, the thickened random variable $\boldsymbol{x}_\delta$ (Definition B.2) satisfies for any $\epsilon > 0$*

$$R_{\delta+\epsilon}(\boldsymbol{x}_\delta) \le R_\epsilon(\boldsymbol{x}). \tag{B.3.10}$$

The proof of Lemma B.6 is deferred to Section B.3.2. Using Lemma B.6, the above program can be realized, because the random variable $\boldsymbol{x}_\delta$ has a density that is uniform with respect to the Lebesgue measure.

(Proof of Theorem 4.1). The upper bound is readily shown. If S is any ϵ-cover of the support of $\boldsymbol{x}$ with cardinality $\mathcal{N}_\epsilon(\operatorname{Supp}(\boldsymbol{x}))$, then consider the coding scheme assigning to each $\boldsymbol{\xi} \in \operatorname{Supp}(\boldsymbol{x})$ the reconstruction $\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}' \in S} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_2$, with ties broken arbitrarily. Then ties occur with probability zero, and the fact that S covers $\operatorname{Supp}(\boldsymbol{x})$ at scale ϵ guarantees distortion no larger than ϵ; the rate of this scheme is $\log_2 \mathcal{N}_\epsilon(\operatorname{Supp}(\boldsymbol{x}))$.

For the lower bound, let $0 < \delta < \frac{1}{2}$, and consider the thickened random variable $\boldsymbol{x}_\delta$. By Lemma B.6, we have

$$R_{\delta+\epsilon}(\boldsymbol{x}_\delta) \le R_\epsilon(\boldsymbol{x}). \tag{B.3.11}$$

Since $\boldsymbol{x}_\delta$ has a Lebesgue density that is uniform, we can then apply the Shannon lower bound, in the form (B.3.5), to get

$$\log_2 \operatorname{vol}(\operatorname{Supp}(\boldsymbol{x}_\delta)) - \log_2 \operatorname{vol}(B_{\delta+\epsilon}) - O(\log D) \le R_\epsilon(\boldsymbol{x}). \tag{B.3.12}$$

Finally, we need to lower bound the ratio

$$\frac{\operatorname{vol}(\operatorname{Supp}(\boldsymbol{x}_\delta))}{\operatorname{vol}(B_{\delta+\epsilon})} \tag{B.3.13}$$

in terms of the covering number. Since $\operatorname{Supp}(\boldsymbol{x}_\delta) = \operatorname{Supp}(\boldsymbol{x}) + B_\delta$, where $+$ here denotes the Minkowski sum, a standard application of volume bound arguments (see e.g. [Ver18, Proposition 4.2.12]) gives

$$\operatorname{vol}(\operatorname{Supp}(\boldsymbol{x}_\delta)) \ge \mathcal{N}_{2\delta}(\operatorname{Supp}(\boldsymbol{x})) \operatorname{vol}(B_\delta). \tag{B.3.14}$$

Hence

$$\frac{\mathrm{vol}(\mathrm{Supp}(\boldsymbol{x}_\delta))}{\mathrm{vol}(B_{\delta+\epsilon})} \geq \mathcal{N}_{2\delta}(\mathrm{Supp}(\boldsymbol{x}))\frac{\mathrm{vol}(B_\delta)}{\mathrm{vol}(B_{\delta+\epsilon})} \tag{B.3.15}$$

$$= \mathcal{N}_{2\delta}(\mathrm{Supp}(\boldsymbol{x}))\left(\frac{\delta}{\delta+\epsilon}\right)^D. \tag{B.3.16}$$

Choosing $\delta = \epsilon/2$ gives from the Shannon lower bound (B.3.12) and the above estimates

$$\log_2 \mathcal{N}_\epsilon(\mathrm{Supp}(\boldsymbol{x})) - O(D) \leq R_\epsilon(\boldsymbol{x}), \tag{B.3.17}$$

as was to be shown.

□

B.3.2 Proof of Lemma B.6

(Proof of Lemma B.6). It suffices to show that any code for $\boldsymbol{x}$ with expected squared distortion ϵ^2 produces a code for $\boldsymbol{x}_\delta$ with the same rate and distortion not much larger, for a suitable choice of δ. So fix such a code for $\boldsymbol{x}$, achieving rate R and expected squared distortion ϵ^2. We write $\hat{\boldsymbol{x}}$ for the reconstructed random variable using this code, and $\mathrm{q} : \mathrm{Supp}(\boldsymbol{x}) \to \mathrm{Supp}(\boldsymbol{x})$ for the associated encoding-decoding mapping (i.e., $\hat{\boldsymbol{x}} = \mathrm{q}(\boldsymbol{x})$).

Now let $\mathcal{S}_k$ denote the k-th hypersphere in the support of $\boldsymbol{x}$. There is an orthonormal basis $\boldsymbol{U}_k \in \mathbb{R}^{D\times d_k}$ such that $\mathrm{Span}(\mathcal{S}_k) = \mathrm{Span}(\boldsymbol{U}_k)$. The following orthogonal decomposition of the support set $\mathcal{S}$ will be used repeatedly throughout the proof. We have

$$\mathcal{S}_\delta = \{\boldsymbol{\xi} \in \mathbb{R}^D \mid \exists k \in [K] \ : \ \mathrm{dist}(\boldsymbol{\xi}, \mathcal{S}_k) \leq \delta\} \tag{B.3.18}$$

$$= \bigcup_{k\in[K]} \{\boldsymbol{\xi} \in \mathbb{R}^D \mid \mathrm{dist}(\boldsymbol{\xi}, \mathcal{S}_k) \leq \delta\}. \tag{B.3.19}$$

By orthogonal projection, for any $k \in [K]$ any $\boldsymbol{\xi} \in \mathbb{R}^D$ can be written as $\boldsymbol{\xi} = \boldsymbol{\xi}^\parallel + \boldsymbol{\xi}^\perp$, with $\boldsymbol{\xi}^\parallel \in \mathrm{Span}(\mathcal{S}_k)$ and $\langle \boldsymbol{\xi}^\parallel, \boldsymbol{\xi}^\perp \rangle = 0$. Then for any $\boldsymbol{\xi}' \in \mathcal{S}_k$, we have

$$\|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_2^2 = \left\|\boldsymbol{\xi}^\parallel + \boldsymbol{\xi}^\perp - \boldsymbol{\xi}'\right\|_2^2 = \left\|\boldsymbol{\xi}^\parallel\right\|_2^2 + \left\|\boldsymbol{\xi}^\perp\right\|_2^2 + \|\boldsymbol{\xi}'\|_2^2 - 2\left\langle \boldsymbol{\xi}^\parallel, \boldsymbol{\xi}' \right\rangle \tag{B.3.20}$$

$$\geq \left\|\boldsymbol{\xi}^\parallel\right\|_2^2 + \|\boldsymbol{\xi}'\|_2^2 - 2\left\langle \boldsymbol{\xi}^\parallel, \boldsymbol{\xi}' \right\rangle \tag{B.3.21}$$

$$= \left\|\boldsymbol{\xi}^\parallel - \boldsymbol{\xi}'\right\|_2^2. \tag{B.3.22}$$

Further, it is known that for any nonzero $\boldsymbol{\xi}^\parallel \in \mathrm{Span}(\mathcal{S}_k)$,

$$\inf_{\boldsymbol{\xi}'\in\mathcal{S}_k} \left\|\boldsymbol{\xi}^\parallel - \boldsymbol{\xi}'\right\|_2^2 = \left\|\boldsymbol{\xi}^\parallel - \frac{\boldsymbol{\xi}^\parallel}{\|\boldsymbol{\xi}^\parallel\|_2}\right\|_2^2. \tag{B.3.23}$$

If $\boldsymbol{\xi}^{\|}$ is zero, it is clear that the above distance is equal to 1 for every $\boldsymbol{\xi}' \in \mathcal{S}_k$. Hence, if we define a projection mapping $\pi_{\mathcal{S}_k}(\boldsymbol{\xi})$ by

$$\pi_{S_k}(\boldsymbol{\xi}) = \frac{\boldsymbol{U}_k \boldsymbol{U}_k^\top \boldsymbol{\xi}}{\|\boldsymbol{U}_k^\top \boldsymbol{\xi}\|_2} \tag{B.3.24}$$

for any $\boldsymbol{\xi} \in \mathbb{R}^D$ with $\boldsymbol{U}_k^\top \boldsymbol{\xi} \neq \mathbf{0}$, then $\pi_{\mathcal{S}_k}(\boldsymbol{\xi}) = \arg\min_{\boldsymbol{\xi}' \in \mathcal{S}_k} \|\boldsymbol{\xi}' - \boldsymbol{\xi}\|_2$. We choose $0 < \delta < 1$, so that the thickened set $\mathcal{S}_\delta$ contains no points $\boldsymbol{\xi} \in \mathbb{R}^D$ at which any of the projection maps $\pi_{\mathcal{S}_k}$ is not well-defined. So the thickened set $\mathcal{S}_\delta$ satisfies

$$\mathcal{S}_\delta = \bigcup_{k \in [K]} \left\{ \boldsymbol{\xi} \in \mathbb{R}^D \;\middle|\; \left\| \boldsymbol{\xi} - \frac{\boldsymbol{U}_k \boldsymbol{U}_k^\top \boldsymbol{\xi}}{\|\boldsymbol{U}_k^\top \boldsymbol{\xi}\|_2} \right\|_2 \leq \delta \right\}. \tag{B.3.25}$$

These distances can be rewritten in terms of the orthogonal decomposition as

$$\left\| \boldsymbol{\xi} - \frac{\boldsymbol{U}_k \boldsymbol{U}_k^\top \boldsymbol{\xi}}{\|\boldsymbol{U}_k^\top \boldsymbol{\xi}\|_2} \right\|_2^2 = \|\boldsymbol{\xi}\|_2^2 - 2\|\boldsymbol{U}_k^\top \boldsymbol{\xi}\|_2 + 1 \tag{B.3.26}$$

$$= \left\| \boldsymbol{\xi}^{\|} \right\|_2^2 + \left\| \boldsymbol{\xi}^{\perp} \right\|_2^2 - 2\|\boldsymbol{U}_k^\top \boldsymbol{\xi}^{\|}\|_2 + 1 \tag{B.3.27}$$

$$= \left\| \boldsymbol{\xi}^{\perp} \right\|_2^2 + \left(\left\| \boldsymbol{\xi}^{\|} \right\|_2 - 1 \right)^2. \tag{B.3.28}$$

We are going to show next that every such $\boldsymbol{\xi} \in \mathcal{S}_\delta$ can be uniquely associated to a projection onto a single subspace in the mixture, which will allow us to define a corresponding projection onto $\mathcal{S}$. Given a $\boldsymbol{\xi} \in \mathcal{S}_\delta$, by the above, we can find a subspace $\boldsymbol{U}_k$ such that the orthogonal decomposition $\boldsymbol{\xi} = \boldsymbol{\xi}_k^{\|} + \boldsymbol{\xi}_k^{\perp}$ satisfies

$$\left\| \boldsymbol{\xi}_k^{\perp} \right\|_2^2 + \left(\left\| \boldsymbol{\xi}_k^{\|} \right\|_2 - 1 \right)^2 \leq \delta^2. \tag{B.3.29}$$

Consider the decomposition $\boldsymbol{\xi} = \boldsymbol{\xi}_j^{\|} + \boldsymbol{\xi}_j^{\perp}$ for some $j \neq k$. We have

$$\left\| \boldsymbol{\xi}_j^{\|} \right\|_2 = \left\| \boldsymbol{U}_j \boldsymbol{U}_j^\top \boldsymbol{\xi} \right\|_2 = \left\| \boldsymbol{U}_j \boldsymbol{U}_j^\top (\boldsymbol{U}_k \boldsymbol{U}_k^\top \boldsymbol{\xi} + (\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^\top)\boldsymbol{\xi}) \right\|_2 \tag{B.3.30}$$

$$= \left\| \boldsymbol{U}_j \boldsymbol{U}_j^\top (\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^\top)\boldsymbol{\xi} \right\|_2 \tag{B.3.31}$$

$$\leq \left\| (\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^\top)\boldsymbol{\xi} \right\|_2 = \left\| \boldsymbol{\xi}_k^{\perp} \right\|_2 \leq \delta, \tag{B.3.32}$$

where the second line uses the orthogonality assumption on the subspaces $\boldsymbol{U}_k$, and the third uses the fact that orthogonal projections are nonexpansive. Hence, the j-th distance satisfies

$$\left\| \boldsymbol{\xi}_j^{\perp} \right\|_2^2 + \left(\left\| \boldsymbol{\xi}_j^{\|} \right\|_2 - 1 \right)^2 \geq (1 - \delta)^2. \tag{B.3.33}$$

This implies that if $0 < \delta < 1/2$, every $\boldsymbol{\xi} \in \mathcal{S}_\delta$ has a unique closest subspace in the mixture. Hence, under this condition, the following mapping $\pi_{\mathcal{S}} : \mathcal{S}_\delta \to \mathcal{S}$ is well-defined:

$$\pi_{\mathcal{S}}(\boldsymbol{\xi}) = \pi_{\mathcal{S}_{k_\star}}(\boldsymbol{\xi}), \;\; \text{where} \;\; k_\star = \operatorname*{arg\,min}_{k \in [K]} \operatorname{dist}(\boldsymbol{\xi}, \mathcal{S}_k). \tag{B.3.34}$$

Now, we define a code for $\boldsymbol{x}_\delta$ by

$$\hat{\boldsymbol{x}}_\delta = \mathrm{q}(\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)). \tag{B.3.35}$$

Clearly this is associated to a rate-R code for $\boldsymbol{x}_\delta$, because it uses the encoding-decoding mappings from the rate-R code for $\boldsymbol{x}$. We have to show that it achieves small distortion. We calculate

$$\mathbb{E}\left[\|\boldsymbol{x}_\delta - \hat{\boldsymbol{x}}_\delta\|_2^2\right] = \mathbb{E}\left[\|\boldsymbol{x}_\delta - \mathrm{q}(\pi_{\mathcal{S}}(\boldsymbol{x}_\delta))\|_2^2\right] \tag{B.3.36}$$

$$\leq \left(\mathbb{E}\left[\|\boldsymbol{x}_\delta - \pi_{\mathcal{S}}(\boldsymbol{x}_\delta)\|_2^2\right]^{1/2} + \mathbb{E}\left[\|\pi_{\mathcal{S}}(\boldsymbol{x}_\delta) - \mathrm{q}(\pi_{\mathcal{S}}(\boldsymbol{x}_\delta))\|_2^2\right]^{1/2}\right)^2, \tag{B.3.37}$$

where the inequality uses the Minkowski inequality. Now, by Definitions B.1 and B.2, we have deterministically

$$\|\boldsymbol{x}_\delta - \pi_{\mathcal{S}}(\boldsymbol{x}_\delta)\|_2^2 \leq \delta^2, \tag{B.3.38}$$

so the expectation also satisfies this estimate. For the second term, it will suffice to characterize the density of the random variable $\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)$ as being sufficiently close to the density of $\boldsymbol{x}$—which, as Assumption B.3 implies, is a mixture of uniform distributions on each sub-sphere $\mathcal{S}_k$. By the argument above, every point $\boldsymbol{\xi} \in \mathcal{S}_\delta$ can be associated to one and only one subspace $\boldsymbol{U}_k$, which means that the mixture components in the definition of $\mathcal{S}_\delta$ (recall Definition B.2) do not overlap. Hence, the density $\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)$ can be characterized by studying the effect of $\pi_{\mathcal{S}_k}$ on the conditional random variable $\boldsymbol{x}_\delta$, conditioned on being drawn from $\mathcal{S}_{k,\delta}$. Denote this measure by $\mu_{k,\delta}$. We claim that the pushforward of this measure under $\pi_{\mathcal{S}_k}$ is uniform on $\mathcal{S}_k$. To see that this holds, we recall Equation (B.3.28), which gives the characterization

$$\mathcal{S}_{k,\delta} = \left\{\boldsymbol{\xi}^\parallel + \boldsymbol{\xi}^\perp \;\middle|\; \boldsymbol{\xi}^\parallel \in \mathrm{Span}(\boldsymbol{U}_k), \boldsymbol{\xi}^\perp \in \mathrm{Span}(\boldsymbol{U}_k)^\perp, \left\|\boldsymbol{\xi}^\perp\right\|_2^2 + \left(\left\|\boldsymbol{\xi}^\parallel\right\|_2 - 1\right)^2 \leq \delta^2\right\}. \tag{B.3.39}$$

The conditional distribution in question is uniform on this set; we need to show that the projection $\pi_{\mathcal{S}_k}$ applied to this conditional random variable yields a random variable that is uniform on $\mathcal{S}_k$. With respect to these coordinates, we have seen that $\pi_{\mathcal{S}_k}(\boldsymbol{\xi}^\parallel + \boldsymbol{\xi}^\perp) = \boldsymbol{\xi}^\parallel / \|\boldsymbol{\xi}^\parallel\|_2$. Hence, for any $\boldsymbol{\xi} \in \mathcal{S}_k$, we have that the preimage of $\boldsymbol{\xi}$ in $\mathcal{S}_{k,\delta}$ under $\pi_{\mathcal{S}_k}$ is

$$\pi_{\mathcal{S}_k}^{-1}(\boldsymbol{\xi}) = \left\{r\boldsymbol{\xi} + \boldsymbol{\xi}^\perp \;\middle|\; r > 0, \boldsymbol{\xi}^\perp \in \mathrm{Span}(\boldsymbol{U}_k)^\perp, \left\|\boldsymbol{\xi}^\perp\right\|_2^2 + (r-1)^2 \leq \delta^2\right\}. \tag{B.3.40}$$

To show that $(\pi_{\mathcal{S}_k})_\sharp \mu_{k,\delta}$ is uniform, we need to decompose the integral of the uniform density on $\mathcal{S}_{k,\delta}$ in a way that makes it clear that each of the fibers $\pi_{\mathcal{S}_k}^{-1}(\boldsymbol{\xi})$ (for each $\boldsymbol{\xi} \in \mathcal{S}_k$) "contributes" equally to the integral.[6] We have by

[6] More rigorously, this corresponds to decomposing the uniform density on $\mathcal{S}_{k,\delta}$ into a regular conditional density corresponding to $\boldsymbol{\xi} \in \mathcal{S}_k$, and showing that the corresponding density on $\boldsymbol{\xi}$ is uniform. The proof makes it clear this is true.

Definition B.2

$$\mathrm{vol}(\mathcal{S}_{k,\delta}) = \iint_{\mathrm{Span}(\boldsymbol{U}_k)\times\mathrm{Span}(\boldsymbol{U}_k)^\perp} \mathbf{1}_{\|\boldsymbol{\xi}^\perp\|_2^2+(\|\boldsymbol{\xi}^\parallel\|_2-1)^2\le\delta^2}\mathrm{d}\boldsymbol{\xi}^\parallel\mathrm{d}\boldsymbol{\xi}^\perp. \tag{B.3.41}$$

In particular, the integration over the orthogonal coordinates factors. Let $\mathrm{d}\boldsymbol{\theta}^d$ denote the uniform (Haar) measure on the sphere of radius 1 in $\mathbb{R}^d$. Converting the $\boldsymbol{\xi}^\parallel$ integral to polar coordinates, we have

$$\mathrm{vol}(\mathcal{S}_{k,\delta}) = \int_{[0,\infty)}\int_{\mathbb{S}^{d_k-1}}\int_{\mathrm{Span}(\boldsymbol{U}_k)^\perp} r^{d_k-1}\mathbf{1}_{\|\boldsymbol{\xi}^\perp\|_2^2+(r-1)^2\le\delta^2}\mathrm{d}r\mathrm{d}\boldsymbol{\theta}^{d_k}\mathrm{d}\boldsymbol{\xi}^\perp. \tag{B.3.42}$$

Comparing to the fiber representation (B.3.40), we see that we need to "integrate out" over the r and $\boldsymbol{\xi}^\perp$ components of the preceding integral in order to verify that the pushforward is uniform. But this is evident, as the previous expression shows that the value of this integral is independent of $\boldsymbol{\xi}^\parallel$—or, equivalently in context, the value of the spherical component $\boldsymbol{\theta}^{d_k}$.

Thus it follows from the above argument that $\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)$ is uniform. Because the assumption on δ implies that the mixture components in the distribution of $\boldsymbol{x}_\delta$ do not overlap, the mixing weights π_k are also preserved in the image $\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)$, and in particular, the distribution of $\pi_{\mathcal{S}}(\boldsymbol{x}_\delta)$ is equal to the distribution of $\boldsymbol{x}$. Hence the second term in Equation (B.3.37) satisfies

$$\mathbb{E}\left[\|\pi_{\mathcal{S}}(\boldsymbol{x}_\delta) - \mathrm{q}(\pi_{\mathcal{S}}(\boldsymbol{x}_\delta))\|_2^2\right] = \mathbb{E}\left[\|\boldsymbol{x} - \mathrm{q}(\boldsymbol{x})\|_2^2\right] \le \epsilon^2, \tag{B.3.43}$$

because q is a distortion-ϵ code for $\boldsymbol{x}$.

We have thus shown that the hypothesized rate-R, (expected squared) distortion-ϵ^2 code for $\boldsymbol{x}$ produces a rate-R, (expected squared) distortion $\delta+\epsilon$ code for $\boldsymbol{x}_\delta$. This establishes that

$$R_{\delta+\epsilon}(\boldsymbol{x}_\delta) \le R_\epsilon(\boldsymbol{x}), \tag{B.3.44}$$

as was to be shown.

□

Bibliography

[AKH15] Yousset I Abdel-Aziz, Hauck Michael Karara, and Michael Hauck. "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry". *Photogrammetric engineering & remote sensing* 81.2 (2015), pp. 103–107.

[AMS09] P-A Absil, R Mahony, and R Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Apr. 2009.

[AAJ+16] Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. "Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization". *SIAM Journal on Optimization* 26.4 (2016), pp. 2775–2799. eprint: https://doi.org/10.1137/140979861.

[AEB06] Michal Aharon, Michael Elad, and Alfred Bruckstein. "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation". *IEEE Transactions on signal processing* 54.11 (2006), pp. 4311–4322.

[ACM12] William K. Allard, Guangliang Chen, and Mauro Maggioni. "Multi-scale geometric methods for data sets II: Geometric Multi-Resolution Analysis". *Applied and Computational Harmonic Analysis* 32.3 (2012), pp. 435–462.

[AR20] Jason M. Allred and Kaushik Roy. "Controlled Forgetting: Targeted Stimulation and Dopaminergic Plasticity Modulation for Unsupervised Lifelong Learning in Spiking Neural Networks". *Frontiers in Neuroscience* 14 (2020).

[ADG+16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent". *Advances in neural information processing systems*. 2016, pp. 3981–3989.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[AGM+15] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. "Simple, Efficient, and Neural Algorithms for Sparse Coding". *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grünwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, July 2015, pp. 113–149.

[AW18] Aharon Azulay and Yair Weiss. "Why do deep convolutional networks generalize so poorly to small image transformations?" *arXiv preprint arXiv:1805.12177* (2018).

[BJC85] B. Ans, J. Hérault, and C. Jutten. "Architectures neuromimétiques adaptatives : Détection de primitives". *Cognitiva* 2 (1985), pp. 593–597.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". *arXiv preprint arXiv:1607.06450* (2016).

[BM24] Hao Bai and Yi Ma. "Improving neuron-level interpretability with white-box language models". *arXiv preprint arXiv:2410.16443* (2024).

[BGN+17] Bowen Baker, Otkrist Gupta, N. Naik, and R. Raskar. "Designing Neural Network Architectures using Reinforcement Learning". *ArXiv* abs/1611.02167 (2017).

[Bal11] Pierre Baldi. "Autoencoders, unsupervised learning and deep architectures". *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*. UTLW'11. Washington, USA: JMLR.org, 2011, pp. 37–50.

[BH89] Pierre Baldi and Kurt Hornik. "Neural networks and principal component analysis: Learning from examples without local minima". *Neural networks* 2.1 (1989), pp. 53–58.

[BS16] Pierre Baldi and Peter Sadowski. "A theory of local learning, the learning channel, and the optimality of backpropagation". *Neural Networks* 83 (2016), pp. 51–74.

[BLZ+22] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models". *arXiv preprint arXiv:2201.06503* (2022).

[BNX+23] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. "All are Worth Words: A ViT Backbone for Diffusion Models". *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22669–22679.

[BSM+20] Pinglei Bao, Liang She, Mason McGill, and Doris Y. Tsao. "A map of object space in primate inferotemporal cortex". *Nature* 583 (2020), pp. 103–108.

[BYL+23] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. "MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation". *International Conference on Machine Learning*. PMLR. 2023, pp. 1737–1752.

[BKS15] Boaz Barak, Jonathan A Kelner, and David Steurer. "Dictionary learning and tensor decomposition via the sum-of-squares method". *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. New York, NY, USA: ACM, June 2015.

[BT09] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[BHM+19] 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 (2019), pp. 15849–15854.

[BBH+15] Elizabeth A Bell, Patrick Boehnke, T Mark Harrison, and Wendy L Mao. "Potentially biogenic carbon preserved in a 4.1 billion-year-old zircon". *Proceedings of the National Academy of Sciences* 112.47 (2015), pp. 14518–14521.

[Bel57] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Ben23] Max Bennett. *A Brief History of Intelligence: Evolution, AI, and the Five Breakthroughs That Made Our Brains*. Mariner Books, 2023.

[BN24a] Jeremy Bernstein and Laker Newhouse. *Old Optimizer, New Norm: An Anthology*. 2024. arXiv: 2409.20325 [cs.LG].

[Bla72] R. Blahut. "Computation of channel capacity and rate-distortion functions". *IEEE Transactions on Information Theory* 18.4 (1972), pp. 460–473.

[BRL+23] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. "Align your latents: High-resolution video synthesis with latent diffusion models". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 22563–22575.

[BAV25] Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. "How to build a consistency model: Learning flow maps via self-distillation". *arXiv preprint arXiv:2505.18825* (2025).

[BBF+01] Lorna Booth, Jehoshua Bruck, M. Franceschetti, and Ronald Meester. "Covering Algorithms, Continuum Percolation and the Geometry of Wireless Networks". *Ann. Appl. Probab.* 13 (July 2001).

[Bor97] Vivek S Borkar. "Stochastic approximation with two time scales". *Systems & Control Letters* 29.5 (1997), pp. 291–294.

[BDS16] Vivek S Borkar, Raaz Dwivedi, and Neeraja Sahasrabudhe. "Gaussian approximations in high dimensional estimation". *Systems & Control Letters* 92 (2016), pp. 42–45.

[Bos50] R. Boscovich. *De calculo probabilitatum que respondent diversis valoribus summe errorum post plures observationes, quarum singule possient esse erronee certa quadam quantitate*. 1750.

[Bou23] Nicolas Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, Mar. 2023.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BN24b] Arwen Bradley and Preetum Nakkiran. "Classifier-Free Guidance is a Predictor-Corrector". *arXiv [cs.LG]* (Aug. 2024). arXiv: 2408.09000 [cs.LG].

[BN20] Guy Bresler and Dheeraj Nagaraj. "Sharp representation theorems for relu networks with precise dependence on depth". *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20 Article 897. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 10697–10706.

[BB11] Haim Brezis and Haim Brézis. *Functional analysis, Sobolev spaces and partial differential equations*. Vol. 2. 3. Springer, 2011.

[BEJ25] Paige Bright, Alan Edelman, and Steven G Johnson. "Matrix Calculus (for Machine Learning and Beyond)". *arXiv preprint arXiv:2501.14787* (2025).

[BDS19] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". *International Conference on Learning Representations (ICLR)*. 2019.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language models are few-shot learners". *arXiv preprint arXiv:2005.14165* (2020).

[BDE+24] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. "Genie: Generative interactive environments". *Forty-first International Conference on Machine Learning*. 2024.

[BM13] Joan Bruna and Stéphane Mallat. "Invariant Scattering Convolution Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1872–1886.

[BGW21] Sam Buchanan, Dar Gilboa, and John Wright. "Deep Networks and the Multiple Manifold Problem". *International Conference on Learning Representations*. 2021.

[BPM+25] Sam Buchanan, Druv Pai, Yi Ma, and Valentin De Bortoli. "On the edge of memorization in diffusion models". *arXiv preprint arXiv:2508.17689* (2025).

[CD91] M. Frank Callier and A. Charles Desoer. *Linear System Theory*. Springer-Verlag, 1991.

[Can06] E. Candès. "Compressive sampling". *Proceedings of the International Congress of Mathematicians*. 2006.

[CT05a] E. Candès and T. Tao. "Decoding by linear programming". *IEEE Transactions on Information Theory* 51.12 (2005).

[CT05b] E. Candès and T. Tao. "Error Correction via Linear Programming". *IEEE Symposium on FOCS* (2005), pp. 295–308.

[CMM+21] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. 2021. arXiv: 2006.09882 [cs.CV].

[CTM+21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers". *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[Cha66] Gregory J. Chaitin. "On the Length of Programs for Computing Finite Binary Sequences". *J. ACM* 13.4 (Oct. 1966), pp. 547–569.

[CYY+22] Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. "ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction". *Journal of Machine Learning Research* 23.114 (2022), pp. 1–103.

[CJG+15] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. "PCANet: A simple deep learning baseline for image classification?" *TIP* (2015).

[CFG+15] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. "Shapenet: An information-rich 3d model repository". *arXiv preprint arXiv:1512.03012* (2015).

[CWM+17] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. "Deep adaptive image clustering". *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5879–5887.

[CT17] Le Chang and Doris Tsao. "The Code for Facial Identity in the Primate Brain". *Cell* 169 (June 2017), 1013–1028.e14.

[CMP+21] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. "An attentive survey of attention models". *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.5 (2021), pp. 1–32.

[CRE+19] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. "On tiny episodic memories in continual learning". *arXiv preprint arXiv:1902.10486* (2019).

[CXE+24] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. "LaRa: Efficient Large-Baseline Radiance Fields". *European Conference on Computer Vision (ECCV)*. 2024.

[CHZ+23] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. "Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data". *International Conference on Machine Learning*. PMLR. 2023, pp. 4672–4712.

[CRB+18] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. "Neural ordinary differential equations". *Advances in neural information processing systems* 31 (2018).

[CCL+23] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. "Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions". *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[CZG+24] Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. "Exploring low-dimensional subspace in diffusion models for controllable image editing". *Advances in Neural Information Processing Systems* 37 (2024), pp. 27340–27371.

[CZL+25] Siyi Chen, Yimeng Zhang, Sijia Liu, and Qing Qu. "The Dual Power of Interpretable Token Embeddings: Jailbreaking Attacks and Defenses for Diffusion Model Unlearning". *arXiv preprint arXiv:2504.21307* (2025).

[CKN+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". *arXiv preprint arXiv:2002.05709* (2020).

[CLH+24] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. "Symbolic discovery of optimization algorithms". *Advances in neural information processing systems* 36 (2024).

[CAP20] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. "Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion". *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 6968–6979.

[Cho17] Francois Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807.

[CKM+23] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. “Diffusion Posterior Sampling for General Noisy Inverse Problems”. *The Eleventh International Conference on Learning Representations*. 2023.

[CW16a] Taco Cohen and Max Welling. “Group equivariant convolutional networks”. *International Conference on Machine Learning*. 2016, pp. 2990–2999.

[CW16b] Taco Cohen and Max Welling. “Group equivariant convolutional networks”. *International conference on machine learning*. PMLR. 2016, pp. 2990–2999.

[CW16c] Taco S. Cohen and Max Welling. “Group Equivariant Convolutional Networks”. *CoRR* abs/1602.07576 (2016). arXiv: 1602.07576.

[CV95] Corinna Cortes and Vladimir Vapnik. “Support-Vector Networks”. *Mach. Learn.* 20.3 (1995), pp. 273–297.

[CT91] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[Cov64] Thomas Cover. “Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition”. *IEEE TRANSACTIONS ON ELECTRONIC COMPUTERS* (1964).

[Cyb89] George V. Cybenko. “Approximation by superpositions of a sigmoidal function”. *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.

[D D00] D. Donoho. “High-dimensional data analysis: The curses and blessings of dimensionality”. *AMS Math Challenges Lecture* (2000).

[DM03] D. Donoho and M. Elad. “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization”. *PNAS* 100.5 (2003), pp. 2197–2202.

[DTL+22] Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, and Yi Ma. “CTRL: Closed-Loop Transcription to an LDR via Minimaxing Rate Reduction”. *Entropy* 24.4 (2022).

[Dan02] George B Dantzig. “Linear Programming”. *Operations Research* 50.1 (2002), pp. 42–47.

[DSD+23a] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. "Ambient Diffusion: Learning Clean Distributions from Corrupted Data". *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[DSD+23b] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. "Ambient Diffusion: Learning Clean Distributions from Corrupted Data". *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 288–313.

[DGG+25] Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. "Distributional Diffusion Models with Scoring Rules". *arXiv preprint arXiv:2502.02483* (2025).

[DCM+23] Aaron Defazio, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. "Optimal linear decay learning rate schedules and further refinements". *arXiv preprint arXiv:2310.07831* (2023).

[DLW+23] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. "Objaverse-XL: a universe of 10M+ 3D objects". *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023.

[DDS22] Julie Delon, Agnes Desolneux, and Antoine Salmona. "Gromov–Wasserstein distances between Gaussian distributions". *Journal of Applied Probability* 59.4 (2022), pp. 1178–1198.

[DDS+09a] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". *CVPR09*. 2009.

[DDS+09b] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[DCL+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[DN21a] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021.

[DN21b] Prafulla Dhariwal and Alexander Quinn Nichol. "Diffusion Models Beat GANs on Image Synthesis". *Advances in Neural Information Processing Systems*. Ed. by A Beygelzimer, Y Dauphin, P Liang, and J Wortman Vaughan. 2021.

[Don01] D L Donoho. "Sparse components of images and optimal atomic decompositions". *Constructive approximation* 17.3 (Jan. 2001), pp. 353–382.

[DVD+98] D L Donoho, M Vetterli, R A DeVore, and I Daubechies. "Data compression and harmonic analysis". *IEEE transactions on information theory / Professional Technical Group on Information Theory* 44.6 (Oct. 1998), pp. 2435–2476.

[Don05] David L Donoho. "Neighborly polytopes and sparse solutions of underdetermined linear equations". *Stanford Technical Report 2005-04* (2005).

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[DFK+22] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. "Google scanned objects: A high-quality dataset of 3d scanned household items". *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2553–2560.

[DSC22] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. "Activation functions in deep learning: A comprehensive survey and benchmark". *Neurocomputing* 503 (2022), pp. 92–108.

[DSK17] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. "A Learned Representation for Artistic Style". *arXiv preprint arXiv:1610.07629* (2017).

[EY36] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". *Psychometrika* 1.3 (Sept. 1936), pp. 211–218.

[EAS98] A Edelman, T Arias, and S Smith. "The Geometry of Algorithms with Orthogonality Constraints". *SIAM Journal on Matrix Analysis and Applications* 20.2 (Jan. 1998), pp. 303–353.

[EA06] Michael Elad and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries". *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society* 15.12 (Dec. 2006), pp. 3736–3745.

[ELP+97] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. "The farthest point strategy for progressive image sampling". *IEEE transactions on image processing* 6.9 (1997), pp. 1305–1315.

[EHO+22a] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. "Toy models of superposition". *arXiv preprint arXiv:2209.10652* (2022).

[EHO+22b] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. "Toy Models of Superposition". *Transformer Circuits Thread* (2022).

[ETT+17] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "A rotation and a translation suffice: Fooling CNNs with simple transformations". *arXiv preprint arXiv:1712.02779* (2017).

[EKB+24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. "Scaling rectified flow transformers for high-resolution image synthesis". *Forty-first international conference on machine learning*. 2024.

[ERO21] Patrick Esser, Robin Rombach, and Bjorn Ommer. "Taming transformers for high-resolution image synthesis". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12873–12883.

[EGW+10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. *The PASCAL Visual Object Classes (VOC) Challenge*. 2010. arXiv: 0909.5206 [cs.CV].

[FPH+25] Tyler Farghly, Peter Potaptchik, Samuel Howard, George Deligiannidis, and Jakiw Pidstrigach. "Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive". *arXiv preprint arXiv:2510.02305* (2025).

[FZS22] William Fedus, Barret Zoph, and Noam Shazeer. "Switch transformers: scaling to trillion parameter models with simple and efficient sparsity". *J. Mach. Learn. Res.* 23.1 (Jan. 2022).

[Fel49] William Feller. "On the Theory of Stochastic Processes, with Particular Reference to Applications". 1949.

[FCR20] Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. "Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study". *International Conference on Machine Learning*. PMLR. 2020, pp. 3133–3144.

[FCR19] Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. "Convergence of learning dynamics in stackelberg games". *arXiv preprint arXiv:1906.01217* (2019).

[Fuk69] Kunihiko Fukushima. "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements". *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), pp. 322–333.

[Fuk80] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics* 36 (1980), pp. 193–202.

[GIF+23] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. *DataComp: In search of the next generation of multimodal datasets*. 2023. arXiv: 2304.14108 [cs.CV].

[GTT+25] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. "Scaling and evaluating sparse autoencoders". *The Thirteenth International Conference on Learning Representations*. 2025.

[GHH+24] Ruiqi Gao, Aleksander Hoły'nski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. "CAT3D: Create Anything in 3D with Multi-View Diffusion Models". *Advances in Neural Information Processing Systems (NeurIPS)*. Oral. 2024.

[GG23] Guillaume Garrigos and Robert M Gower. "Handbook of convergence theorems for (stochastic) gradient methods". *arXiv preprint arXiv:2301.11235* (2023).

[GWX+25] Zheng Geng, Nan Wang, Shaocong Xu, Chongjie Ye, Bohan Li, Zhaoxi Chen, Sida Peng, and Hao Zhao. "One View, Many Worlds: Single-Image to 3D Object Meets Generative Domain Randomization for One-Shot 6D Pose Estimation". *arXiv preprint arXiv:2509.07978* (2025).

[GDB+25] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. "Mean flows for one-step generative modeling". *arXiv preprint arXiv:2505.13447* (2025).

[GVM25] Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. "Denoising score matching with random features: Insights on diffusion models from precise learning curves". *arXiv preprint arXiv:2502.00336* (2025).

[GRS+23] Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. "Looped transformers as programmable computers". *International Conference on Machine Learning*. PMLR. 2023, pp. 11398–11442.

[Gil61] E. N. Gilbert. "Random Plane Networks". *Journal of the Society for Industrial and Applied Mathematics* 9.4 (1961), pp. 533–543. eprint: https://doi.org/10.1137/0109045.

[GC19] Aaron Gokaslan and Vanya Cohen. *OpenWebText Corpus*. http://Skylion007.github.io/OpenWebTextCorpus. 2019.

[GPM+14a] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014.

[GPM+14b] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[GDG+17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. "Accurate, large minibatch sgd: Training imagenet in 1 hour". *arXiv preprint arXiv:1706.02677* (2017).

[GWT+23] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. "Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives". *arXiv preprint arXiv:2311.18259* (2023).

[GL10] Karol Gregor and Yann LeCun. "Learning fast approximations of sparse coding". *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010, pp. 399–406.

[Gri11] Rémi Gribonval. "Should Penalized Least Squares Regression be Interpreted as Maximum A Posteriori Estimation?" *IEEE transactions on signal processing: a publication of the IEEE Signal Processing Society* 59.5 (May 2011), pp. 2405–2410.

[GJB15] Remi Gribonval, Rodolphe Jenatton, and Francis Bach. "Sparse and spurious: Dictionary learning with noise and outliers". *IEEE transactions on information theory* 61.11 (Nov. 2015), pp. 6298–6319.

[Gro87] Stephen Grossberg. "Competitive Learning: From Interactive Activation to Adaptive Resonance". *Cogn. Sci.* 11 (1987), pp. 23–63.

[GD24] Albert Gu and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces". *First conference on language modeling*. 2024.

[GDP+23] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. "On memorization in diffusion models". *arXiv preprint arXiv:2310.02664* (2023).

[GYR+23] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning". *arXiv preprint arXiv:2307.04725* (2023).

[HCL06] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping". *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*. 2006, pp. 1735–1742.

[HY01] M.H. Hansen and B. Yu. "Model Selection and the Principle of Minimum Description Length". *Journal of American Statistical Association* 96 (2001), pp. 746–774.

[Har15] Yuval Noah Harari. *Sapiens: A Brief History of Humankind*. Harper, 2015.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Second. Springer, 2009.

[Haw21] Jeff Hawkins. *A Thousand Brains: A New Theory of Intelligence*. Basic Books, 2021.

[HCX+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked autoencoders are scalable vision learners". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16000–16009.

[HFW+19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". *arXiv preprint arXiv:1911.05722* (2019).

[HZR+16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[HZR+16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[Hea+56] Thomas Little Heath et al. *The thirteen books of Euclid's Elements*. Courier Corporation, 1956.

[HRU+17a] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[HRU+17b] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". *Advances in neural information processing systems* 30 (2017).

[HS06] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". *Science* 313.5786 (2006), pp. 504–507. eprint: https://science.sciencemag.org/content/313/5786/504.full.pdf.

[HZ93] Geoffrey E. Hinton and Richard S. Zemel. "Autoencoders, minimum description length and Helmholtz free energy". *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1993, pp. 3–10.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 6840–6851.

[HS21] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[HS22a] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". *arXiv [cs.LG]* (July 2022). arXiv: 2207.12598 [cs.LG].

[HS22b] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2022.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". *Neural computation* 9 (Dec. 1997), pp. 1735–80.

[HSD20] David Hong, Yue Sheng, and Edgar Dobriban. *Selecting the number of components in PCA via random signflips*. 2020. arXiv: 2012.02985 [math.ST].

[HZG+23] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. "Lrm: Large reconstruction model for single image to 3d". *arXiv preprint arXiv:2311.04400* (2023).

[Hot33] H. Hotelling. "Analysis of a Complex of Statistical Variables into Principal Components". *Journal of Educational Psychology* (1933).

[HZM+23] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. "SugarCrepe: Fixing Hackable Benchmarks for Vision-Language Compositionality". *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2023.

[HDZ+25] Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. "CUPID: Generative 3D Reconstruction via Joint Object and Pose Modeling". *arXiv preprint arXiv:2510.20776* (2025).

[HYH+22] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. "Capturing and Inferring Dense Full-Body Human-Scene Contact". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13274–13285.

[HLV+17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely Connected Convolutional Networks". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269.

[HM99] Jinggang Huang and David Mumford. "Statistics of natural images and models". *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 1. IEEE. 1999, pp. 541–547.

[HW59] D.H. Hubel and T.N. Wiesel. "eceptive fields of single neurones in the cat's striate cortex". *J. Physiol.* 148.3 (1959), pp. 574–591.

[HCS+24] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. "Sparse Autoencoders Find Highly Interpretable Features in Language Models". *The Twelfth International Conference on Learning Representations*. 2024.

[HKV19] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, eds. *Automatic Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[HKJ+21] Uiwon Hwang, Heeseung Kim, Dahuin Jung, Hyemi Jang, Hyungyu Lee, and Sungroh Yoon. "Stein Latent Optimization for Generative Adversarial Networks". *arXiv preprint arXiv:2106.05319* (2021).

[Hyv05] Aapo Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". *Journal of Machine Learning Research* 6.24 (2005), pp. 695–709.

[HO97] Aapo Hyvärinen and Erkki Oja. "A Fast Fixed-Point Algorithm for Independent Component Analysis". *Neural Computation* 9.7 (1997), pp. 1483–1492.

[HO00a] Aapo Hyvärinen and Erkki Oja. "Independent Component Analysis: Algorithms and Applications". *Neural Networks* 13.4-5 (2000), pp. 411–430.

[HO00b] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". *Neural Networks* 13.4 (2000), pp. 411–430.

[IWW+21] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. *OpenCLIP*. Version 0.1. If you use this software, please cite it as below. July 2021.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *ICML*. 2015, pp. 448–456.

[JGB+21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. "Perceiver: General Perception with Iterative Attention". *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4651–4664.

[JAD+21] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. "Robust Compressed Sensing MRI with Deep Generative Priors". *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 14938–14954.

[Jam15] G J O Jameson. "A simple proof of Stirling's formula for the gamma function". *The Mathematical Gazette* 99.544 (Mar. 2015), pp. 68–74.

[JRR+24] Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. "On the Origins of Linear Representations in Large Language Models". *International Conference on Machine Learning*. Vol. 235. PMLR, 2024, pp. 21879–21911.

[JJT+24] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. "Lvsm: A large view synthesis model with minimal 3d inductive bias". *arXiv preprint arXiv:2410.17242* (2024).

[Jol02] I. Jollife. *Principal Component Analysis*. 2nd. Springer-Verlag, 2002.

[Jol86] I. Jolliffe. *Principal Component Analysis*. New York, NY: Springer-Verlag, 1986.

[Jon82] Douglas Samuel Jones. *The theory of generalised functions*. Cambridge University Press, 1982.

[JJB+] Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. "Muon: An optimizer for hidden layers in neural networks, 2024". *URL https://kellerjordan.github. io/posts/muon* 6 ().

[JT20] Sheena A. Josselyn and Susumu Tonegawa. "Memory engrams: Recalling the past and imagining the future". *Science* 367 (2020).

[KS21] Z Kadkhodaie and E P Simoncelli. "Stochastic solutions for linear inverse problems using the prior implicit in a denoiser". *Adv. Neural Information Processing Systems (NeurIPS)*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021.

[Kal60] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems" (1960).

[KG24] Mason Kamb and Surya Ganguli. "An analytic theory of creativity in convolutional diffusion models". *arXiv preprint arXiv:2412.20292* (2024).

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling Laws for Neural Language Models". *arXiv preprint arXiv:2001.08361* (2020).

[Kar22a] Andrej Karpathy. *nanoGPT*. https://github.com/karpathy/nanoGPT. 2022.

[Kar22b] Andrej Karpathy. *The spelled-out intro to neural networks and backpropagation: building micrograd*. YouTube. Aug. 16, 2022. URL: https://www.youtube.com/watch?v=VMj-3S1tku0 (visited on 08/17/2025).

[KK18] Ronald Kemker and Christopher Kanan. "FearNet: Brain-Inspired Model for Incremental Learning". *International Conference on Learning Representations*. 2018.

[KKL+23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.* 42.4 (2023), pp. 139–1.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014).

[KW13a] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". *arXiv [stat.ML]* (Dec. 2013). arXiv: 1312.6114v11 [stat.ML].

[KW13b] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114* (2013).

[KW19] Diederik P Kingma and Max Welling. "An introduction to variational autoencoders". *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[KPR+17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. "Overcoming catastrophic forgetting in neural networks". *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[KUM+17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. "Self-normalizing neural networks". *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 972–981.

[Kli11] Ronald Kline. "Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence". *IEEE Annals of the History of Computing* 33.4 (2011), pp. 5–16.

[KTV18] Artemy Kolchinsky, Brendan D Tracey, and Steven Van Kuyk. "Caveats for information bottleneck in deterministic scenarios". *arXiv preprint arXiv:1808.07593* (2018).

[Kol98] Andrei N. Kolmogorov. "On Tables of Random Numbers (Reprinted from "Sankhya: The Indian Journal of Statistics", Series A, Vol. 25 Part 4, 1963)". *Theor. Comput. Sci.* 207 (1998), pp. 387–395.

[KS12] Irwin Kra and Santiago R Simanca. "On Circulant Matrices". *Notices of the American Mathematical Society* 59 (2012), pp. 368–377.

[Kra91] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". *AIChE Journal* 37.2 (1991), pp. 233–243.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images" (2009).

[KNH14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "The CIFAR-10 dataset". *online: http://www.cs.toronto.edu/kriz/cifar.html* 55 (2014).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[KZZ+23] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. "Multi-concept customization of text-to-image diffusion". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 1931–1941.

[Lab24] Black Forest Labs. *FLUX*. https://github.com/black-forest-labs/flux. 2024.

[LBB+25] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. "FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space". *arXiv [cs.GR]* (June 2025). arXiv: 2506.15742 [cs.GR].

[LRM+12] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. "Building high-level features using large scale unsupervised learning". *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML'12. Edinburgh, Scotland: Omnipress, 2012, pp. 507–514.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation* 1.4 (1989), pp. 541–551.

[LBB+98a] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LBB+98b] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LCH+06] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. "A Tutorial on Energy-Based Learning". Jan. 2006.

[LPM03] Ann Lee, Kim Pedersen, and David Mumford. "The Nonlinear Statistics of High-Contrast Patches in Natural Images". *International Journal of Computer Vision* 54 (Aug. 2003).

[LSJ+16] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. "Gradient descent only converges to minimizers". *Conference on learning theory*. PMLR. 2016, pp. 1246–1257.

[Lee02] John M. Lee. "Introduction to Smooth Manifolds". 2002.

[LMH+18] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. "Noise2Noise: Learning Image Restoration without Clean Data". *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2965–2974.

[LY24] Gen Li and Yuling Yan. "O (d/T) convergence theory for diffusion probabilistic models under minimal assumptions". *arXiv preprint arXiv:2409.18959* (2024).

[LFD+22] Haochuan Li, Farzan Farnia, Subhro Das, and Ali Jadbabaie. "On convergence of gradient descent ascent: A tight local analysis". *International Conference on Machine Learning*. PMLR. 2022, pp. 12717–12740.

[Li17] Xi-Lin Li. "Preconditioned stochastic gradient descent". *IEEE transactions on neural networks and learning systems* 29.5 (2017), pp. 1454–1466.

[LTP25] Qinyu Li, Yee Whye Teh, and Razvan Pascanu. "NoProp: Training Neural Networks without Back-propagation or Forward-propagation". *arXiv preprint arXiv:2503.24322* (2025).

[LH25] Tianhong Li and Kaiming He. "Back to basics: Let denoising generative models denoise". *arXiv preprint arXiv:2511.13720* (2025).

[LZQ24] Wenda Li, Huijie Zhang, and Qing Qu. "Shallow diffuse: Robust and invisible watermarking through low-dimensional subspaces in diffusion models". *arXiv preprint arXiv:2410.21088* (2024).

[LWQ25] Xiang Li, Rongrong Wang, and Qing Qu. "Towards Understanding the Mechanisms of Classifier-Free Guidance". *arXiv [cs.CV]* (May 2025). arXiv: 2505.19210 [cs.CV].

[LB19] Yanjun Li and Yoram Bresler. "Multichannel sparse blind deconvolution on the sphere". *IEEE Transactions on Information Theory* 65.11 (2019), pp. 7415–7436.

[LCW+24] Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. "Photomaker: Customizing realistic human photos via stacked id embedding". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 8640–8650.

[LRZ+12] Xiao Liang, Xiang Ren, Zhengdong Zhang, and Yi Ma. "Repairing Sparse Low-Rank Texture". *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 482–495.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[LZ94] T Linder and R Zamir. "On the asymptotic tightness of the Shannon lower bound". *IEEE transactions on information theory* 40.6 (1994), pp. 2026–2031.

[Lin88] R. Linsker. "Self-organization in a perceptual network". *Computer* 21.3 (1988), pp. 105–117.

[LCB+23] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. "Flow Matching for Generative Modeling". *The Eleventh International Conference on Learning Representations*. 2023.

[LMR17] Anna V. Little, Mauro Maggioni, and Lorenzo Rosasco. "Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature". *Applied and Computational Harmonic Analysis* 43.3 (2017), pp. 504–567.

[LMZ+24] Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. "Infini-gram: Scaling unbounded n-gram language models to a trillion tokens". *arXiv preprint arXiv:2401.17377* (2024).

[LSY+25] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. "Muon is Scalable for LLM Training". *arXiv preprint arXiv:2502.16982* (2025).

[LV09] Z. Liu and L. Vandenberghe. "Semidefinite programming methods for system realization and identification". *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009, pp. 4676–4681.

[LV10] Zhang. Liu and Lieven. Vandenberghe. "Interior-Point Method for Nuclear Norm Approximation with Application to System Identification". *SIAM Journal on Matrix Analysis and Applications* 31.3 (2010), pp. 1235–1256. eprint: https://doi.org/10.1137/090755436.

[LMR+15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. "SMPL: A skinned multi-person linear model". *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–16.

[LC98] William E. Lorensen and Harvey E. Cline. "Marching cubes: a high resolution 3D surface construction algorithm". *Seminal Graphics: Pioneering Efforts That Shaped the Field, Volume 1*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 347–353.

[LH17] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". *International Conference on Learning Representations*. 2017.

[LH19] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". *arXiv preprint arXiv:1711.05101* (2019).

[MYP+10] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. "Generalized power method for sparse principal component analysis". *Journal of Machine Learning Research* 11 (2010), pp. 517–553.

[MDH+07a] Y. Ma, H. Derksen, W. Hong, and J. Wright. "Segmentation of multivariate mixed data via lossy coding and compression". *To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence* (2007).

[MKS+04] Y. Ma, J. Košecká, S. Soatto, and S. Sastry. *An Invitation to 3-D Vision, From Images to Models*. New York: Springer-Verlag, 2004.

[MDH+07b] Yi Ma, Harm Derksen, Wei Hong, and John Wright. "Segmentation of multivariate mixed data via lossy data coding and compression". *IEEE transactions on pattern analysis and machine intelligence* 29.9 (2007), pp. 1546–1562.

[MHN13] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier nonlinearities improve neural network acoustic models". *Proc. ICML*. Vol. 30. Citeseer. 2013, p. 3.

[MGT+19] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. "AMASS: Archive of motion capture as surface shapes". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5442–5451.

[MBP14] Julien Mairal, Francis Bach, and Jean Ponce. "Sparse Modeling for Image and Vision Processing". *Foundations and Trends® in Computer Graphics and Vision* 8.2-3 (2014), pp. 85–283.

[MSM93] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a Large Annotated Corpus of English: The Penn T reebank". *Computational Linguistics* 19.2 (1993). Ed. by Julia Hirschberg, pp. 313–330.

[Mar06] Andrei Andreevich Markov. "An example of statistical investigation of the text Eugene Onegin concerning the connection of samples in chains". *Science in Context* 19.4 (2006), pp. 591–600.

[MBD+21] James Martens, Andy Ballard, Guillaume Desjardins, Grzegorz Swirszcz, Valentin Dalibard, Jascha Sohl-Dickstein, and Samuel S Schoenholz. "Rapid training of deep neural networks without skip connections or normalization layers using Deep Kernel Shaping". *arXiv [cs.LG]* (Oct. 2021). arXiv: 2110.01765 [cs.LG].

[MS19] David McAllester and Karl Stratos. *Formal Limitations on the Measurement of Mutual Information*. 2019.

[MMR+06] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence: August 31, 1955". *AI Mag.* 27.4 (Dec. 2006), pp. 12–14.

[MC89] Michael McCloskey and Neal J Cohen. "Catastrophic interference in connectionist networks: The sequential learning problem". *Psychology of learning and motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[MP43] Warren McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133.

[MM70] Jerry M. Mendel and Robert W. Mclaren. "Reinforcement-learning control and pattern recognition systems". *In Mendel, J. M. and Fu, K. S., editors, Adaptive, Learning and Pattern Recognition Systems: Theory and Applications* (1970), pp. 287–318.

[MXB+16] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. *Pointer Sentinel Mixture Models*. 2016. arXiv: 1609.07843 [cs.CL].

[MM12] Stephan Mertens and Cristopher Moore. "Continuum percolation thresholds in two dimensions". *Phys. Rev. E* 86 (6 Dec. 2012), p. 061109.

[MON+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. "Occupancy Networks: Learning 3D Reconstruction in Function Space". *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4455–4465.

[MSS+22] Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafal Bogacz, and Thomas Lukasiewicz. "Predictive coding: Towards a future of deep learning beyond backpropagation?" *arXiv preprint arXiv:2202.09467* (2022).

[Min54] Marvin Minsky. "Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem". PhD thesis. Princeton Univerity, 1954.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. the MIT Press, 1969.

[Mir60] L Mirsky. "SYMMETRIC GAUGE FUNCTIONS AND UNITARILY INVARIANT NORMS". *The Quarterly Journal of Mathematics* 11.1 (Jan. 1960), pp. 50–59.

[MCS+22] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. "Autosdf: Shape priors for 3d completion, reconstruction and generation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 306–315.

[Miy61] K. Miyasawa. "An empirical bayes estimator of the mean of a normal population". *Bull. Inst. Internat. Statist.* 38 (1961).

[MKK+18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. "Spectral normalization for generative adversarial networks". *arXiv preprint arXiv:1802.05957* (2018).

[MRY+11] Hossein Mobahi, Shankar Rao, Allen Yang, Shankar Sastry, and Yi Ma. "Segmentation of Natural Images by Texture and Boundary Compression". *the International Journal of Computer Vision* 95.1 (2011), pp. 86–98.

[MLE19] Vishal Monga, Yuelong Li, and Yonina C Eldar. "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing". *arXiv preprint arXiv:1912.10557* (2019).

[MÁM+24] Edmund RR Moody, Sandra Álvarez-Carretero, Tara A Mahendrarajah, James W Clark, Holly C Betts, Nina Dombrowski, Lénárd L Szánthó, Richard A Boyle, Stuart Daines, Xi Chen, et al. "The nature of the last universal common ancestor and its impact on the early Earth system". *Nature Ecology & Evolution* 8.9 (2024), pp. 1654–1666.

[MKW+22] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. "SLIP: Self-supervision Meets Language-Image Pre-training". *Computer Vision – ECCV 2022*. Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Cham: Springer Nature Switzerland, 2022, pp. 529–544.

[MKH19] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. "When does label smoothing help?" *Advances in neural information processing systems* 32 (2019).

[Mum96] David Mumford. "The Statistical Description of Visual Signals". 1996.

[MG99] David Mumford and Basilis Gidas. "Stochastic Models for Generic Images". *Quarterly of Applied Mathematics* 59 (July 1999).

[MK07] Joseph F Murray and Kenneth Kreutz-Delgado. "Learning sparse overcomplete codes for images". *The Journal of VLSI Signal Processing Systems for Signal Image and Video Technology* 46.1 (Mar. 2007), pp. 1–13.

[MLS94] R. Murray, Zexiang Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Florida: CRC Press, 1994.

[NDE+13] S. Nam, M.E. Davies, M. Elad, and R. Gribonval. "The cosparse analysis model and algorithms". *Applied and Computational Harmonic Analysis* 34.1 (2013), pp. 30–56.

[NGE+20] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. "Polygen: An autoregressive generative model of 3d meshes". *International conference on machine learning*. PMLR. 2020, pp. 7220–7229.

[NMR44] John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944.

[Neu58] John von Neumann. *The computer and the brain*. USA: Yale University Press, 1958.

[NJD+22] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. "Point-e: A system for generating 3d point clouds from complex prompts". *arXiv preprint arXiv:2212.08751* (2022).

[ND21] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved Denoising Diffusion Probabilistic Models". *International Conference on Machine Learning (ICML)*. 2021.

[NZM+24] Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. "Towards a Mechanistic Explanation of Diffusion Model Generalization". *arXiv preprint arXiv:2411.19339* (2024).

[NMM19] Oliver Nina, Jamison Moody, and Clarissa Milligan. "A Decoder-Free Approach for Unsupervised Clustering and Manifold Learning with Random Triplet Mining". *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE Computer Society. 2019, pp. 3987–3994.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[NIG+18] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. "Activation functions: Comparison of trends in practice and research for deep learning". *arXiv preprint arXiv:1811.03378* (2018).

[Oja82] Erkki Oja. "A simplified neuron model as a principal component analyzer". *Journal of Mathematical Biology* 15 (1982), pp. 267–273.

[OLC+25] Kazusato Oko, Licong Lin, Yuhang Cai, and Song Mei. "A statistical theory of contrastive pre-training and multimodal generative AI". *arXiv [cs.LG]* (Jan. 2025). arXiv: 2501.04641 [cs.LG].

[OF97] B A Olshausen and D J Field. "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision research* 37.23 (Dec. 1997), pp. 3311–3325.

[OF96] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". *Nature* 381 (June 1996), p. 607.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". *arXiv [cs.LG]* (July 2018). arXiv: 1807.03748 [cs.LG].

[OVK17] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning". *arXiv [cs.LG]* (Nov. 2017). arXiv: 1711.00937 [cs.LG].

[Ope24] OpenAI. *Sora: Creating video from text*. https://openai.com/sora. 2024.

[ODM+23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "Dinov2: Learning robust visual features without supervision". *arXiv preprint arXiv:2304.07193* (2023).

[ODM+24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV].

[PBW+24] Druv Pai, Sam Buchanan, Ziyang Wu, Yaodong Yu, and Yi Ma. "Masked Completion via Structured Diffusion with White-Box Transformers". *The Twelfth International Conference on Learning Representations*. 2024.

[PPC+23] Druv Pai, Michael Psenka, Chih-Yuan Chiu, Manxi Wu, Edgar Dobriban, and Yi Ma. "Pursuit of a discriminative representation for multiple subspaces via sequential games". *Journal of the Franklin Institute* 360.6 (2023), pp. 4135–4171.

[PCY+23] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. "Aria Digital Twin: A New Benchmark Dataset for Egocentric 3D Machine Perception". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 20133–20143.

[PKL+16] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. *The LAMBADA dataset: Word prediction requiring a broad discourse context*. 2016. arXiv: 1606.06031 [cs.CL].

[PHD20] Vardan Papyan, XY Han, and David L Donoho. "Prevalence of Neural Collapse during the terminal phase of deep learning training". *arXiv preprint arXiv:2008.08186* (2020).

[PRE17] Vardan Papyan, Yaniv Romano, and Michael Elad. "Convolutional neural networks analyzed via convolutional sparse coding". *The Journal of Machine Learning Research* 18.1 (2017), pp. 2887–2938.

[PCV24] Kiho Park, Yo Joong Choe, and Victor Veitch. "The Linear Representation Hypothesis and the Geometry of Large Language Models". *International Conference on Machine Learning*. PMLR. 2024, pp. 39643–39666.

[Par04] Andrew Parker. *In The Blink Of An Eye: How Vision Sparked The Big Bang Of Evolution*. Basic Books, 2004.

[PSR+24] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. "Reconstructing Hands in 3D with Transformers". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

[Pea09] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[Pea01] K. Pearson. "On Lines and Planes of Closest Fit to Systems of Points in Space". *Philosophical Magazine* 2.6 (1901), pp. 559–572.

[PX23] William Peebles and Saining Xie. "Scalable diffusion models with transformers". *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4195–4205.

[PV25] Liangzu Peng and René Vidal. "Mathematics of continual learning". *arXiv preprint arXiv:2504.17963* (2025).

[PNM+20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. "Convolutional Occupancy Networks". *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 523–540.

[PSV+18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. "FiLM: Visual Reasoning with a General Conditioning Layer". *AAAI Conference on Artificial Intelligence*. 2018.

[PRR+22] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. "A self-supervised descriptor for image copy detection". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14532–14542.

[Pla99] S. E. Plamer. *Vision Science: Photons to Phenomenology*. The MIT Press, 1999.

[PEL+23] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. "Sdxl: Improving latent diffusion models for high-resolution image synthesis". *arXiv preprint arXiv:2307.01952* (2023).

[PW22] Yuri Poliyanski and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2022.

[PPR+24] Michael Psenka, Druv Pai, Vishal Raman, Shankar Sastry, and Yi Ma. "Representation Learning via Manifold Flattening and Reconstruction". *Journal of Machine Learning Research* 25.132 (2024), pp. 1–47.

[QSM+17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[QYS+17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[QLZ19] Qing Qu, Xiao Li, and Zhihui Zhu. "A nonconvex approach for exact and efficient multichannel sparse blind deconvolution". *Advances in Neural Information Processing Systems*. 2019, pp. 4017–4028.

[QLZ20] Qing Qu, Xiao Li, and Zhihui Zhu. "Exact Recovery of Multichannel Sparse Blind Deconvolution via Gradient Descent". *SIAM Journal on Imaging Sciences* 13.3 (2020), pp. 1630–1652.

[QZL+20a] Qing Qu, Yuexiang Zhai, Xiao Li, Yuqian Zhang, and Zhihui Zhu. "Geometric Analysis of Nonconvex Optimization Landscapes for Overcomplete Learning". *International Conference on Learning Representations*. 2020.

[QZL+20b] Qing Qu, Zhihui Zhu, Xiao Li, Manolis C. Tsakiris, John Wright, and René Vidal. *Finding the Sparsest Vectors in a Subspace: Theory, Algorithms, and Applications*. 2020. arXiv: 2001.06970 [cs.LG].

[RD03] R. Basri and D. Jacobs. "Lambertian reflectance and linear subspaces". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.2 (2003), pp. 218–233.

[RKH+21a] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural L anguage Supervision". *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763.

[RKH+21b] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". *International Conference on Machine Learning (ICML)*. 2021.

[RKH+21c] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763.

[RMC16] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". *arXiv preprint arXiv:1511.06434* (2016). arXiv: 1511.06434 [cs.LG].

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". *OpenAI blog* 1.8 (2019), p. 9.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[RDN+22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical Text-Conditional Image Generation with CLIP Latents". *arXiv [cs.CV]* (Apr. 2022). arXiv: 2204.06125 [cs.CV].

[RPC+06] Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann Cun. "Efficient Learning of Sparse Representations with an Energy-Based Model". *Advances in Neural Information Processing Systems*. Ed. by B Schölkopf, J Platt, and T Hoffman. Vol. 19. MIT Press, 2006.

[RS11] M Raphan and E P Simoncelli. "Least squares estimation without priors or supervision". *Neural Computation* 23.2 (2011). Published online, Nov 2010., pp. 374–420.

[RKS+17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. "iCaRL: Incremental classifier and representation learning". *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010.

[RAG+24] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. "Lossy image compression with foundation diffusion models". *European Conference on Computer Vision*. Springer. 2024, pp. 303–319.

[RBK18] Erwin Riegler, Helmut Bolcskei, and Gunther Koliander. "Rate-distortion theory for general sets and measures". *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, June 2018, pp. 101–105.

[RKB23] Erwin Riegler, Günther Koliander, and Helmut Bölcskei. "Lossy compression of general random variables". *Information and inference: a journal of the IMA* 12.3 (Apr. 2023), pp. 1759–1829.

[Ris78] J. Rissanen. "Paper: Modeling by shortest data description". *Automatica* 14.5 (1978), pp. 465–471.

[Rob56] Herbert E. Robbins. "An Empirical Bayes Approach to Statistics". 1956.

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-Resolution Image Synthesis with Latent Diffusion M odels". *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10674–10685.

[RTB17] Javier Romero, Dimitrios Tzionas, and Michael J Black. "Embodied hands: modeling and capturing hands and bodies together". *ACM Transactions on Graphics (TOG)* 36.6 (2017), pp. 1–17.

[RFB15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[RFB15b] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[Ros57] Frank Rosenblatt. *The perceptron: A perceiving and recognizing automaton*. Report. Project PARA, Cornell Aeronautical Laboratory, Jan. 1957.

[RRD+23] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. "Solving Linear Inverse Problems Provably via Posterior Sampling with Latent Diffusion Models". *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[Row97] Sam T. Roweis. "EM Algorithms for PCA and SPCA". *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*. The MIT Press, 1997, pp. 626–632.

[RAL+24] François Rozet, Gérôme Andry, Francois Lanusse, and Gilles Louppe. "Learning Diffusion Priors from Observations by Expectation Maximization". *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

[RE14] R. Rubinstein and M. Elad. "Dictionary Learning for Analysis-Synthesis Thresholding". *IEEE Transactions on Signal Processing* 62.22 (2014), pp. 5962–5972.

[RLJ+23] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 22500–22510.

[RHW86a] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning internal representations by error propagation". *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[RHW86b] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". *Nature* 323.6088 (Oct. 1986), pp. 533–536.

[SCS+22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". *NeurIPS*. 2022.

[SGZ+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. "Improved Techniques for Training GANs". *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016.

[Sas99] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[SDL+25] Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. "Reasoning with latent thoughts: On the power of looped transformers". *arXiv preprint arXiv:2502.17416* (2025).

[SHT+25] Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Simsekli, and Francis Bach. "The Surprising Agreement Between Convex Optimization Theory and Learning-Rate Scheduling for Large Model Training". *arXiv preprint arXiv:2501.18965* (2025).

[SMB10] Dominik Scherer, Andreas Müller, and Sven Behnke. "Evaluation of pooling operations in convolutional architectures for object recognition". *International conference on artificial neural networks*. Springer. 2010, pp. 92–101.

[Sch14] Denise Schmandt-Besserat. "The evolution of writing". *International encyclopedia of social and behavioral sciences* (2014), pp. 1–15.

[SBV+22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. *LAION-5B: An open large-scale dataset for training next generation image-text models*. 2022. arXiv: 2210.08402 [cs.CV].

[SBZ+25] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. "Flashattention-3: Fast and accurate attention with asynchrony and low-precision". *Advances in Neural Information Processing Systems* 37 (2025), pp. 68658–68685.

[Sha48] C. E. Shannon. "A mathematical theory of communication". *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[Sha59] Claude E Shannon. "Coding theorems for a discrete source with a fidelity criterion". *IRE Nat. Conv. Rec* 4.142-163 (1959), p. 1.

[SMM+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". *ICLR*. 2017.

[SPX+22] Zifan Shi, Sida Peng, Yinghao Xu, Andreas Geiger, Yiyi Liao, and Yujun Shen. "Deep generative models on 3d representations: A survey". *arXiv preprint arXiv:2210.15663* (2022).

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". *arXiv preprint arXiv:1409.1556* (2014).

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". *International Conference on Learning Representations*. 2015.

[SWM+15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 2256–2265.

[SKZ+24] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. "Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency". *The Twelfth International Conference on Learning Representations*. 2024.

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models". *arXiv preprint arXiv:2010.02502* (2020).

[SKC+25] Kiwhan Song, Jaeyeon Kim, Sitan Chen, Yilun Du, Sham Kakade, and Vincent Sitzmann. "Selective Underfitting in Diffusion Models". *arXiv preprint arXiv:2510.01378* (2025).

[SE19] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d 'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[SSX+22] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models". *International Conference on Learning Representations*. 2022.

[SSK+21] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[SWW12] Daniel A Spielman, Huan Wang, and John Wright. "Exact Recovery of Sparsely-Used Dictionaries". *Proceedings of the 25th Annual Conference on Learning Theory*. Ed. by Shie Mannor, Nathan Srebro, and Robert C Williamson. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, 2012, pp. 37.1–37.18.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[SQW15] Ju Sun, Qing Qu, and John Wright. "When are nonconvex problems not scary?" *arXiv preprint arXiv:1510.06096* (2015).

[SQW17a] Ju Sun, Qing Qu, and John Wright. "Complete Dictionary Recovery Over the Sphere I: Overview and the Geometric Picture". *IEEE Transactions on Information Theory* 63.2 (2017), pp. 853–884.

[SQW17b] Ju Sun, Qing Qu, and John Wright. "Complete dictionary recovery over the sphere I: Overview and the geometric picture". *IEEE Transactions on Information Theory* 63.2 (2017), pp. 853–884.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 1–9.

[TZL+25] Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, et al. "Kimi Linear: An Expressive, Efficient Attention Architecture". *arXiv preprint arXiv:2510.26692* (2025).

[Tea] Moonshot Team.

[Tel16] Matus Telgarsky. "Benefits of depth in neural networks". *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 2016, pp. 1517–1539.

[Til15] Andreas M Tillmann. "On the computational intractability of exact and approximate dictionary learning". *IEEE signal processing letters* 22.1 (Jan. 2015), pp. 45–49.

[TB99] M. Tipping and C. Bishop. "Probabilistic principal component analysis". *Journal of Royal Statistical Society: Series B* 61.3 (1999), pp. 611–622.

[TZ15] Naftali Tishby and Noga Zaslavsky. "Deep learning and the information bottleneck principle". *2015 IEEE Information Theory Workshop (ITW)*. IEEE. 2015, pp. 1–5.

[TDC+24] Shengbang Tong, Xili Dai, Yubei Chen, Mingyang Li, ZENGYI LI, Brent Yi, Yann LeCun, and Yi Ma. "Unsupervised Learning of Structured Representation via Closed-Loop Transcription". *Conference on Parsimony and Learning*. Ed. by Yuejie Chi, Gintare Karolina Dziugaite, Qing Qu, Atlas Wang Wang, and Zhihui Zhu. Vol. 234. Proceedings of Machine Learning Research. PMLR, Jan. 2024, pp. 440–457.

[TDW+23] Shengbang Tong, Xili Dai, Ziyang Wu, Mingyang Li, Brent Yi, and Yi Ma. "Incremental Learning of Structured Memory via Closed-Loop Transcription". *The Eleventh International Conference on Learning Representations*. 2023.

[TCD+20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. "Training data-efficient image transformers & distillation through attention. arXiv 2020". *arXiv preprint arXiv:2012.12877* 2.3 (2020).

[Tu07] Zhuowen Tu. "Learning Generative Models via Discriminative Approaches". *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8.

[Tur50] Alan Turing. "Computing Machinary and Intelligence". *Mind* 59 (1950), pp. 433–460.

[Tur36] Alan M. Turing. "On Computable Numbers, with an Application to the E ntscheidungsproblem". *Proceedings of the London Mathematical Society* 2.42 (1936), pp. 230–265.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Instance normalization: The missing ingredient for fast stylization". *arXiv preprint arXiv:1607.08022* (2016).

[Ume02] Shinji Umeyama. "Least-squares estimation of transformation parameters between two point patterns". *IEEE Transactions on pattern analysis and machine intelligence* 13.4 (2002), pp. 376–380.

[VM96] P. Van Overschee and B. de Moor. *Subspace Identification for Linear Systems*. Kluwer Academic, 1996.

[VSP+17a] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[VSP+17b] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". *Advances in neural information processing systems* 30 (2017).

[VST+20] Gido M Ven, Hava T Siegelmann, Andreas S Tolias, et al. "Brain-inspired replay for continual learning with artificial neural networks". *Nature Communications* 11.1 (2020), pp. 1–14.

[VJO+21] Luca Venturi, Samy Jelassi, Tristan Ozuch, and Joan Bruna. "Depth separation beyond radial functions". *Journal of machine learning research: JMLR* 23 (Feb. 2021), 122:1–122:56. eprint: 2102.01621.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge University Press, 2018.

[VM04] R. Vidal and Y. Ma. "A unified algebraic approach to 2-D and 3-D motion segmentation". *Proceedings of the European Conference on Computer Vision*. 2004.

[VMS16] Rene Vidal, Yi Ma, and S. S. Sastry. *Generalized Principal Component Analysis*. 1st. Springer Publishing Company, Incorporated, 2016.

[VMS05] Rene Vidal, Yi Ma, and Shankar Sastry. "Generalized principal component analysis". *IEEE transactions on pattern analysis and machine intelligence* 27.12 (2005), pp. 1945–1959.

[Vin11] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". *Neural Computation* 23.7 (2011), pp. 1661–1674.

[WWG+12] Andrew Wagner, John Wright, Arvind Ganesh, Zihan Zhou, Hossein Mobahi, and Yi Ma. "Toward a practical face recognition system: Robust alignment and illumination by sparse representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.2 (2012), pp. 372–386.

[WB68] C. Wallace and D. Boulton. "An Information Measure for Classification". *The Computer Journal* 11 (1968), pp. 185–194.

[WD99] C. Wallace and D. Dowe. "Minimum message length and Kolmogorov complexity". *The Computer Journal* 42.4 (1999), pp. 270–283.

[WF65] Marion Dwain Waltz and King-Sun Fu. "A heuristic approach to reinforcement learning control systems". *IEEE Transactions on Automatic Control* 10 (1965), pp. 390–398.

[WVM+25] Deng Wang, Jean Vannier, José M Martín-Durán, María Herranz, and Chiyang Yu. "Preservation and early evolution of scalidophoran ventral nerve cord". *Science Advances* 11.2 (2025), eadr0896.

[WCK+25] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. "Vggt: Visual geometry grounded transformer". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 5294–5306.

[WLP+24] Peng Wang, Huikang Liu, Druv Pai, Yaodong Yu, Zhihui Zhu, Qing Qu, and Yi Ma. "A Global Geometric Analysis of Maximal Coding Rate Reduction". *Forty-first International Conference on Machine Learning*. 2024.

[WLY+25] Peng Wang, Yifu Lu, Yaodong Yu, Druv Pai, Qing Qu, and Yi Ma. "Attention-Only Transformers via Unrolled Subspace Denoising". *Forty-second International Conference on Machine Learning*. 2025.

[WZZ+24] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. "Diffusion Models Learn Low-Dimensional Distributions via Subspace Clustering". *arXiv preprint arXiv:2409.02426* (2024).

[WBW+24] Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. "Instantid: Zero-shot identity-preserving generation in seconds". *arXiv preprint arXiv:2401.07519* (2024).

[WXD+25] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. "Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 5261–5271.

[WI20] Tongzhou Wang and Phillip Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere". *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020.

[WGY+23] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. "Cut and learn for unsupervised object detection and instance segmentation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 3124–3134.

[Wer74] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD Thesis, Applied Mathematics Dept., Harvard Univ., 1974.

[Wer94] Paul J. Werbos. "The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting". 1994.

[WB18] T. Wiatowski and H. Bölcskei. "A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction". *IEEE Transactions on Information Theory* (2018).

[Wie42] Norbert Wiener. "The interpolation, extrapolation and smoothing of stationary time series". *Report of the Services 19, Research Project DIC-6037 MIT* (1942).

[Wie48] Norbert Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. the MIT Press, 1948.

[Wie49] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York: Wiley, 1949.

[Wie61] Norbert Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. 2nd ed. the MIT Press, 1961.

[WM22] John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional M odels: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[WTL+08] John Wright, Yangyu Tao, Zhouchen Lin, Yi Ma, and Heung-Yeung Shum. "Classification via minimum incremental coding length (MICL)". *Advances in Neural Information Processing Systems*. 2008, pp. 1633–1640.

[WYG+09] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. "Robust Face Recognition via Sparse Representation". *IEEE Trans. Pattern Anal. Mach. Intell.* 31.2 (Feb. 2009), pp. 210–227.

[WX20] Denny Wu and J. Xu. "On the Optimal Weighted ℓ_2 Regularization in Overparameterized Linear Regression". *ArXiv* abs/2006.05800 (2020).

[WLW+19] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. "Deep comprehensive correlation mining for image clustering". *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8150–8159.

[WTN+23] Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. "Practical and Asymptotically Exact Conditional Sampling in Diffusion Models". *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[WZG+25] Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. "Amodal3r: Amodal 3d reconstruction from occluded 2d images". *arXiv preprint arXiv:2503.13439* (2025).

[WCL+24] Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. "Theoretical Insights for Diffusion Guidance: A Case Study for Gaussian Mixture Models". *arXiv [cs.LG]* (Mar. 2024). arXiv: 2403.01639 [cs.LG].

[WH20] Yuxin Wu and Kaiming He. "Group Normalization". *International Journal of Computer Vision* 128.3 (2020). Originally arXiv:1803.08494, 2018, pp. 742–755.

[WDL+25] Ziyang Wu, Tianjiao Ding, Yifu Lu, Druv Pai, Jingyuan Zhang, Weida Wang, Yaodong Yu, Yi Ma, and Benjamin David Haeffele. "Token Statistics Transformer: Linear-Time Attention via Variational Rate Reduction". *The Thirteenth International Conference on Learning Representations*. 2025.

[WZP+25] Ziyang Wu, Jingyuan Zhang, Druv Pai, XuDong Wang, Chandan Singh, Jianwei Yang, Jianfeng Gao, and Yi Ma. "Simplifying DINO via Coding Rate Regularization". *Forty-second International Conference on Machine Learning*. 2025.

[XLX+25] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. "Structured 3d latents for scalable and versatile 3d generation". *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 21469–21480.

[XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". *International conference on machine learning*. PMLR. 2016, pp. 478–487.

[XGD+17] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5987–5995.

[XWC+15] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. "Empirical evaluation of rectified activations in convolutional network". *arXiv preprint arXiv:1505.00853* (2015).

[XZS+20] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. "A theory of usable information under computational constraints". *arXiv preprint arXiv:2002.10689* (2020).

[XGX+23] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. "ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding". *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 1179–1189.

[YHB+22] Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. "Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer". *arXiv preprint arXiv:2203.03466* (2022).

[YH21] Greg Yang and J E Hu. "Tensor Programs IV: Feature learning in infinite-width neural networks". *International Conference on Machine Learning* 139 (2021). Ed. by Marina Meila and Tong Zhang, pp. 11727–11737.

[YPB16] Jianwei Yang, Devi Parikh, and Dhruv Batra. "Joint unsupervised learning of deep representations and image clusters". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5147–5156.

[YWZ+25] Jingfeng Yang, Ziyang Wu, Yue Zhao, and Yi Ma. *Language-Image Alignment with Fixed Text Encoders*. 2025. arXiv: 2506.04209 [cs.CV].

[YLN+23] Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. "Looped transformers are better at learning learning algorithms". *arXiv preprint arXiv:2311.12424* (2023).

[YKH24] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. "Gated delta networks: Improving mamba2 with delta rule". *arXiv preprint arXiv:2412.06464* (2024).

[YWZ+24] Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. "Parallelizing linear transformers with the delta rule over sequence length". *Advances in neural information processing systems* 37 (2024), pp. 115491–115522.

[YYY+20] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. "Rethinking Bias-Variance Trade-off for Generalization of Neural Networks". *International Conference on Machine Learning*. 2020.

[YZY+25] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. *CAST: Component-Aligned 3D Scene Reconstruction from an RGB Image*. 2025. arXiv: 2502.12894 [cs.CV].

[YWZ+22] Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. "Neural collapse with normalized features: A geometric analysis over the Riemannian manifold". *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 11547–11560.

[YCH+24] Chun-Hsiao Yeh, Ta-Ying Cheng, He-Yen Hsieh, Chuan-En Lin, Yi Ma, Andrew Markham, Niki Trigoni, Hsiang-Tsung Kung, and Yubei Chen. "Gen4gen: Generative data pipeline for generative multi-concept composition". *arXiv preprint arXiv:2402.15504* (2024).

[YZB+23] Brent Yi, Weijia Zeng, Sam Buchanan, and Yi Ma. "Canonical Factors for Hybrid Neural Fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 3414–3426.

[YCK+23] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. "Diffusion probabilistic models generalize when they fail to memorize". *ICML 2023 workshop on structured probabilistic inference {\&} generative modeling*. 2023.

[YLR+16] Chong You, Chun-Guang Li, Daniel P Robinson, and René Vidal. "Oracle based active set algorithm for scalable elastic net subspace clustering". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3928–3937.

[YBP+24] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Hao Bai, Yuexiang Zhai, Benjamin D Haeffele, and Yi Ma. "White-Box Transformers via Sparse Rate Reduction: Compression Is All There Is?" *Journal of Machine Learning Research* 25.300 (2024), pp. 1–128.

[YBP+23] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin Haeffele, and Yi Ma. "White-box transformers via sparse rate reduction". *Advances in Neural Information Processing Systems* 36 (2023).

[YCY+20] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. "Learning Diverse and Discriminative Representations via the Principle of Maximal Coding Rate Reduction". *Advances in neural information processing systems*. 2020.

[YCO+21] Zeyu Yun, Yubei Chen, Bruno A Olshausen, and Yann LeCun. "Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors". *arXiv preprint arXiv:2103.15949* (2021).

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep Sets". *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 3391–3401.

[ZAK24] Moslem Zamani, Hadi Abbaszadehpeivasti, and Etienne de Klerk. "Convergence rate analysis of the gradient descent–ascent method for convex–concave saddle-point problems". *Optimization Methods and Software* 39.5 (2024), pp. 967–989.

[ZDP+25] Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. "RenderFormer: Transformer-based Neural Rendering of Triangle Meshes with Global Illumination". *ACM SIGGRAPH 2025 Conference Papers*. 2025.

[ZMZ+20] Yuexiang Zhai, Hermish Mehta, Zhengyuan Zhou, and Yi Ma. "Understanding l4-based Dictionary Learning: Interpretation, Stability, and Robustness". *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[ZNW22a] Biao Zhang, Matthias Nießner, and Peter Wonka. "3dilg: Irregular latent grids for 3d generative modeling". *Advances in Neural Information Processing Systems* 35 (2022), pp. 21871–21885.

[ZNW22b] Biao Zhang, Matthias Niessner, and Peter Wonka. "3DILG: Irregular Latent Grids for 3D Generative Modeling". *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 21871–21885.

[ZTN+23] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. "3DShape2VecSet: A 3D Shape Representation for Neural Fields and Generative Diffusion Models". *ACM Trans. Graph.* 42.4 (July 2023).

[ZCB+24] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. "Improving diffusion inverse problem solving with decoupled noise annealing". *arXiv [cs.LG]* (July 2024). arXiv: 2407.01521 [cs.LG].

[ZBH+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". *International Conference on Learning Representations*. 2017.

[ZZL+24] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. "The Emergence of Reproducibility and Consistency in Diffusion Models". *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, 2024, pp. 60558–60590.

[ZRA23a] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. "Adding Conditional Control to Text-to-Image Diffusion Models". *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 3813–3824.

[ZRA23b] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models". *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 3836–3847.

[ZIE+18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. "The unreasonable effectiveness of deep features as a perceptual metric". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.

[ZLL+25] Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. "Generalization of Diffusion Models Arises with a Balanced Representation Space". *arXiv preprint arXiv:2512.20963* (2025).

[ZLG+10] Zhengdong Zhang, Xiao Liang, Arvind Ganesh, and Yi Ma. "TILT: Transform Invariant Low-Rank Textures". *International Journal of Computer Vision* 99 (2010), pp. 1–24.

[ZLC+23a] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. "Michelangelo: Conditional 3D Shape Generation based on Shape-Image-Text Aligned Latent Representation". *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[ZLC+23b] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang Yu, and Shenghua Gao. "Michelangelo: Conditional 3D Shape Generation based on Shape-Image-Text Aligned Latent Representation". *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 73969–73982.

[ZFZ19] Hao Zheng, Faming Fang, and Guixu Zhang. "Cascaded Dilated Dense Network with Two-step Data Consistency for MRI Reconstruction". *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d 'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[ZCZ+25] Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy Feng, Caifeng Zou, Yu Sun, Nikola Borislavov Kovachki, Zachary E Ross, Katherine Bouman, and Yisong Yue. "InverseBench: Benchmarking Plug-and-Play Diffusion Priors for Inverse Problems in Physical Sciences". *The Thirteenth International Conference on Learning Representations*. 2025.

[ZLG+] Ziyan Zheng, Chin Wa Lau, Nian Guo, Xiang Shi, and Shao-Lun Huang. "White-box error correction code transformer". *The Second Conference on Parsimony and Learning (Proceedings Track)*.

[ZZ20] Bo Zhou and S. Kevin Zhou. " DuDoRNet: Learning a Dual-Domain Recurrent Network for Fast MRI Reconstruction With Deep T1 Prior ". *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2020, pp. 4272–4281.

[ZM97a] Song Chun Zhu and David Mumford. "Prior Learning and Gibbs Reaction-Diffusion". *IEEE Trans. Pattern Anal. Mach. Intell.* 19.11 (1997), pp. 1236–1250.

[ZWM97] Song Chun Zhu, Ying Nian Wu, and David Mumford. "Minimax Entropy Principle and Its Application to Texture Modeling". *Neural Computation* 9.8 (1997), pp. 1627–1660.

[ZM97b] Song-Chun Zhu and David Mumford. "Learning generic prior models for visual computation". *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1997), pp. 463–469.

[ZDZ+21] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. "A geometric analysis of neural collapse with unconstrained features". *Advances in Neural Information Processing Systems* 34 (2021), pp. 29820–29834.

[ZL17] Barret Zoph and Quoc V. Le. "Neural Architecture Search with Reinforcement Learning". 2017.