周报

【关于长线短线思考】

长线思维赚钱逻辑是平均线的上升趋势
短线思维赚钱逻辑是平均线上下波动的套利

长线逻辑会剥离噪音，需要清晰的知道标的的价值。
不同于短线思维，短线思维的核心逻辑在价格上，长线思维的核心逻辑在价值上。

什么是价值？
价值是某样物品的价格值多少钱，比如：一根雪糕的价格是 10 元，A 商店卖到 15 元，B 商店卖到 9 元，10，15，9 都是价格。这跟雪糕的价值是 10 元，那么这个 10 从哪里来的呢？也就是说，这个雪糕真正的定价就是价值。
10 元的定价取决于很多的因素：
厂家生成的成本、市场的需求、市场的供给、宏观的货币通胀率、商业模式

如何判断一个资产的价值？如何给一个雪糕定价？
趋势线？移动平均线？
同行业的平均价格？
成本价格？ 市场需要？
通胀率？

投资中大家都敬畏市场先生，是因为这个市场中有太多的手在拉扯价格的走向，这就是为什么没有人能够准确的预测未来的走势，也没有人能够只手遮天控制市场，
大家在不断的讨论策略、讨论趋势本质上讨论的都是一个概率问题，
在不断探索能够大概率赚到钱的策略。

我们都在试图了解更多的知识，提升认知，赚认知的钱。
本质上是，我们在通过获取更多的知识，让大脑建立多元模型，模型既是一种投资策略。
让模型帮助我们在市场中赚到钱。

【关于伪装的思考】

我常常觉得，人就是环境的产物。但却不曾细想环境包含了什么。今天想了下，大概就是：天 地 人组成了环境，天大多是一个地方的气候，能见度，湿度，温度。地是指土地，肥沃还是贫瘠，山脉还是平原或是盆地。人是指这个地方生活的人的习惯，悠闲或是紧张，安静或是热闹，有格调或是粗糙。人在某种环境中就会切换成某种状态，比如回到老家，就会自动切换成精神放松，疏于自律和思考学习的模式，而回到北京，基本不需要自我调节就会进入自律状态中。在比如，但在孩子身边时，就会自动切换成妈妈角色，忽略掉很多自我的细节。当独自一人时，就会切换成自我模式，内心就会充满了对于未知事物的探索欲，最近想去跳伞，但转头看看孩子又打消了这个念头，然后再看到跳伞的视频，又想去，反复纠结。

关于美的思考，美的泛概念其实不止停留在视觉，更多的是意念。就像我们认为一个人是美女或帅哥也不是因为他仅仅长得好看，而是这个人整体的 vibe，这个人的性格外貌，言谈举止，穿着品味，甚至身上的味道，指甲的长度所有的细节都决定了 vibe 的整体效果。就像是一部影视作品，画面的构图，光影的搭配，演员的表达，剧情的结构都决定了整个作品的风格。那么什么是美的呢？爱美之心人皆有之，什么是美？

装是美么？为什么要装？装是什么？
装是在特定环境下，对真实自我进行了一定的包装，以适应环境达到某种心理目的。
为什么大家对装如此诟病，因为人们认为装是不真实的，是假的，在大多价值观中，我们讨厌假的，喜欢真实的东西。
并且包装下的灵魂会觉得异常疲惫，所以大家总会说：装的累不累

然而，礼盒也是礼物最重要的部分，一份礼物就是需要礼物本身加礼盒配合才能凸显出新意，少了礼盒的礼物会令人感到心意不够能够完美表达。
女明星好看，但是素颜和没有滤镜的女明星也不会受大众追捧。
即使是黄金，打造成首饰才会更受消费者的青睐。
所以有趣的灵魂很重要，包装也至关重要，这决定了人的整体 vibe。

我曾经常常因为自己在某种状态下，不自觉切换成某种模式的自己而觉得自己虚伪和不真实。但现在想想在某种环境下的我，也是由我内心驱使我表演出来真实的我。是我在不同维度下对自己的解读和诠释。就像大模型在不同的维度空间中会有不同的特征值，每个空间我都是不同的，但也都是我，不同维度空间的解读即是特征，特征值是带有 bias 的。
我小时候很喜欢看一个卡通叫 百变小樱，现在想想应该是小樱在不用环境下能变化成不同的形象角色，但小樱也始终是小樱自己。

【关于动量的思考】

今天继续读 MoCo Paper，对于流动性和能量和动量有一些思考。

流动性这个词我最开始也是最常听到是在金融领域，但是我后来发现，周围的一切都有流动性。
环境具有流动性，东西放在那里一动不动也会有变化。
人具有流动性，人的身体在流动，情绪在流动，能量在流动，状态在流动，认知在流动。
所谓流动其实就是宇宙万物都在运动，运动产生了时间和速度，运动其实就是流动。
流动或者说是运动是宇宙运作的核心。
宏观事物缺乏流动时，就会变得失去活力，一潭死水。微观的元素就会在流动中对其进行分解和转化，让能量再次利用，所谓物质不灭粉碎而已。
忽然理解了上善若水的另一种含义了，之前总觉得，要像水一样，柔软的，利他的，不与他人争高低。
但却忽略了，水是流动的，流动的水拥有速度和质量，也就是说，水是具有巨大能量的。
水的底色是无限强大的能量，并不屑于与其他事物争高低输赢，有一种强者俯视一切的包容，犹如大象看蝼蚁一般。

动量这个词我最早是在物理学中接触的，算法也很简单，p=m· v，m 表示质量，v 表示速度。
但对于动量这个词本身我今天之前是没有好好思考过的。我今天看到这个词，是在读 MoCo Paper 的时候，里面讲到了一种优化模型的策略：Momentum update

然后我又联想到，前几天听小播客，投资人分享的时候也讲到了金融学中股票上涨也有动量的概念。整体梳理一下思路。

动量是指运动的事物具有的能量，或者是具有流动性的事物具有的能量。速度越快，质量越大，具有的能量越大。

金融市场里，大环境一直猛涨，就会有冲上去的惯性。

回头来看，大模型中动量的解读，这个公式是想用 q 的变化速度牵制 k 的变化速度，m 即赋予 k 参数的质量，（1-m）即赋予 q 参数的质量。thetaK 即为 k 参数变化速度，thetaQ 即为 q 参数的变化速度。用 k 参数产生的动量和 q 参数产生的动量的加总表示 k 参数。此处 m 为动量参数。

be different). (**b**): The key representations are sampled from a ***memory bank*** [61]. (**c**): ***MoCo*** encodes the new keys on-the-fly by a momentum-updated encoder, and maintains a queue (not illustrated in this figure) of keys.

3.2. Momentum Contrast

From the above perspective, contrastive learning is a way of building a discrete dictionary on high-dimensional continuous inputs such as images. The dictionary is *dynamic* in the sense that the keys are randomly sampled, and that the key encoder evolves during training. Our hypothesis is that good features can be learned by a *large* dictionary that covers a rich set of negative samples, while the encoder for the dictionary keys is kept as *consistent* as possible despite its evolution. Based on this motivation, we present Momentum Contrast as described next.

Dictionary as a queue. At the core of our approach is maintaining the dictionary as a *queue* of data samples. This allows us to reuse the encoded keys from the immediate preceding mini-batches. The introduction of a queue decouples the dictionary size from the mini-batch size. Our dictionary size can be much larger than a typical mini-batch size, and can be flexibly and independently set as a hyper-parameter.

The samples in the dictionary are progressively replaced. The current mini-batch is enqueued to the dictionary, and the oldest mini-batch in the queue is removed. The dictionary always represents a sampled subset of all data, while the extra computation of maintaining this dictionary is manageable. Moreover, removing the oldest mini-batch can be beneficial, because its encoded keys are the most outdated and thus the least consistent with the newest ones.

Momentum update. Using a queue can make the dictionary large, but it also makes it intractable to update the key encoder by back-propagation (the gradient should propagate to all samples in the queue). A naïve solution is to copy the key encoder f_k from the query encoder f_q, ignoring this gradient. But this solution yields poor results in experiments (Sec. 4.1). We hypothesize that such failure is caused by the rapidly changing encoder that reduces the key representations' consistency. We propose a momentum update to address this issue.

Formally, denoting the parameters of f_k as θ_k and those of f_q as θ_q, we update θ_k by:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q. \tag{2}$$

Here $m \in [0, 1)$ is a momentum coefficient. Only the parameters θ_q are updated by back-propagation. The momentum update in Eqn.(2) makes θ_k evolve more smoothly than θ_q. As a result, though the keys in the queue are encoded by different encoders (in different mini-batches), the difference among these encoders can be made small. In experiments, a relatively large momentum (*e.g.*, $m = 0.999$, our default) works much better than a smaller value (*e.g.*, $m = 0.9$), suggesting that a slowly evolving key encoder is a core to making use of a queue.

Relations to previous mechanisms. MoCo is a general mechanism for using contrastive losses. We compare it with two existing general mechanisms in Figure 2. They exhibit different properties on the dictionary size and consistency.

The ***end-to-end*** update by back-propagation is a natural mechanism (*e.g.*, [29, 46, 36, 63, 2, 35], Figure 2a). It uses samples in the current mini-batch as the dictionary, so the keys are consistently encoded (by the same set of encoder parameters). But the dictionary size is coupled with the mini-batch size, limited by the GPU memory size. It is also challenged by large mini-batch optimization [25]. Some recent methods [46, 36, 2] are based on pretext tasks driven by local positions, where the dictionary size can be made larger by multiple positions. But these pretext tasks may require special network designs such as patchifying the input [46] or customizing the receptive field size [2], which may complicate the transfer of these networks to downstream tasks.

Another mechanism is the ***memory bank*** approach proposed by [61] (Figure 2b). A memory bank consists of the representations of all samples in the dataset. The dictionary for each mini-batch is randomly sampled from the memory bank with no back-propagation, so it can support a large dictionary size. However, the representation of a sample in