
Review report on the manuscript entitled
“Reinforcement Learning for Options
on Target Volatility Funds”

Introduction

The Authors deal with a continuous dynamical control problem related to the management of a target volatility strategy (TVS) through a target volatility option (TVO). The study is articulated in two phases. In the first one, the Authors derive a closed form solution for a European TVO in two different ways. In the second phase, the Authors apply Reinforcement Learning (RL) to the aforementioned continuous dynamical control problem, even with some element of novelties.

Generally, the results presented in the manuscript are interesting, so, in principle, I am favourable to the publication of the manuscript. Anyway, some revisions have to be made according to the observations below.

Major remarks

- Both in **Abstract** and in **Introduction**, the presentation of the problem to deal with is not well structured. As a matter of fact, first the TVS is briefly introduced, then options and TVOs are only hinted, and lastly the problem is shortly described, without all these topics are well linked each other. I suggest the Authors to improve this presentation.
- The Authors address the investigated continuous stochastic control problem through two different approaches: an exact one providing the analytical solution of the problem, and an approximate one based on RL. As for the previous point, even in this one the connections between the two solution approaches are not entirely intelligible. In other terms, it is not entirely clear how the two approaches relate each other. Is the approximate one necessary given that the exact approach provides an analytical solution to the problem? What is the “utility” to use an

approximate approach given the applicability of an exact one? I suggest the Authors to clarify this issue.

- In Subsection 3.3, three (so-called intuitive) strategies are introduced, S_A , S_B and S_C . No rationales are provided concerning their choices. I suggest the Author to give some details about this point.
- In Section 4, the Authors state that «[t]he first [RL] method is a specific algorithm developed by us to fit the problem [...].» As known, an important feature of any RL method is its convergence to the (unknown) best policy, if any. The Authors do not spend words about this characteristic of the specific algorithm they developed and applied, so I ask them to give some detail about this topic.
- In Subsection 5.1, the Author claim that an appropriate choice of the learning rate starting value constitutes «*a good choice in terms of [...] avoiding over-fitting*». As known, when training a FFNN, standard methods for avoid over-fitting are based on error metrics calculated over validation sets or through K -fold validation approaches. So, it is not clear as the Authors can guarantee no over-fitting without exploiting such methods or similar. I suggest the Authors to carefully clarify this issue.
- So, it is not clear as the Authors can guarantee no overfitting without exploiting this methods or similar.

Minor remarks

- Keywords are poor. Several important “objects” considered in the paper are not listed there as, for instance, the target volatility options, the continuous stochastic control problem, the stochastic differential equations and so on. I suggest the Authors to make the keywords more representative of the contents of the manuscript. The same suggestion holds for the title of the manuscript, in which only the RL solution approach is reported.
 - Along the manuscript, the acronym TVS (target volatility strategy) is often improperly used for indicating a TVS-based portfolio. I suggest the Authors to modify wherever necessary.
 - In Section 4, a feed forward neural network (FFNN) is used for parametrizing the agent policy. In this regard, it is not entirely clear at which stage of the RL-based process the FFNN acts. I suggest the Authors to make clearer this point.
 - Further observations and suggestions are reported on the annotated copy of the manuscript.
-

Reinforcement Learning for Options on Target Volatility Funds*

Roberto Daluiso¹, Emanuele Nastasi², Andrea Pallavicini¹, and Stefano Polo³

¹ Financial Engineering, Intesa Sanpaolo Group,
Largo Mattioli 3, 20121 Milan, Italy

{Roberto.Daluiso, Andrea.Pallavicini}@intesasanpaolo.com

² Financial Engineering, Marketz S.p.A.,
Corso Europa, 2, 20122 Milan, Italy
Emanuele.Nastasi@marketz.eu

³ Financial Engineering, X-Numeris S.r.l.,
Via Turro, 7, 20127 Milan, Italy
Stefano.Polo@xnumeris.com

→ of its

Abstract. We deal with the funding costs rising from hedging the risky securities underlying a target volatility strategy (TVS), a portfolio of risky assets and a risk-free one dynamically rebalanced in order to keep the realized volatility of the portfolio on a certain level. The uncertainty in the TVS risky portfolio composition along with the difference in hedging costs for each component requires to solve a control problem to evaluate the option prices. We derive an analytical solution of the problem in the Black and Scholes (BS) scenario. Then we use Reinforcement Learning (RL) techniques to determine the fund composition leading to the most conservative price under the local volatility (LV) model, for which an *a priori* solution is not available. We show how the performances of the RL agents are compatible with those obtained by applying path-wise the BS analytical strategy to the TVS dynamics, which therefore appears competitive also in the LV scenario.

Keywords. Reinforcement learning, target volatility strategy, hedging costs.

M.S.C. classification. 65C05, 68T07, 91G20.

J.E.L. classification. C63, C45, G13.

Who is meant?

* Supported by ISCRA from Cineca, iscra@cineca.it.

1 Introduction

In the recent years portfolio managers are exposed to very low interest rates and quickly changing market volatilities. An effective solution to control risks under such an environment is given by target volatility strategies (TVSs) (also known as constant volatility targeting), which are able to preserve the portfolio at a predetermined level of volatility. A TVS is a portfolio of risky assets (typically equities) and a risk-free asset dynamically re-balanced with the aim of maintaining the overall portfolio volatility level closed to some target value. The constant volatility approach can help investors to obtain desired risk exposures over the short and long term and it increases the risk-adjusted performance of the portfolio.

These products were initially offered in the Asian markets, see for instance the reports of [3] and [26], which highlight the pros and cons for investors, to be adopted in the following years in many other markets in North America and Europe as depicted in [19]. At the present day we can observe some new market indices based on the mechanism of the target volatility strategy such as Dow Jones Volatility Control Index, and S&P 500 Risk Control Index.

In the recent literature, TVSs are tested to investigate their performances in term of realized returns, see for instance [15] and [20], and the soundness of the volatility targeting algorithm, as described in [16]. Moreover, in the present day, in the derivative market new structured financial products that see TVSs as underlying are becoming widely used by practitioners. Those derivative instruments are known as target volatility options (TVOs) and have gained great interest in the derivative pricing literature, see in particular [1], [6], [7], and [11].

This paper is the first attempt, to the best of our knowledge, to study the funding costs coming from hedging the risky assets underlying the target volatility strategy. In particular, we consider the point of view of a bank selling a call option to a portfolio manager as protection on the capital invested in a TVS. The portfolio manager has the freedom of changing the relative weights of the risky assets during the life of the TVS. Since the risky assets have different hedging costs, the bank shall adjust the price of the protection to include them in the worst-case scenario, i.e. the most expensive strategy in term of the financing costs. Hence, the pricing problem becomes a continuous dynamical control problem over the risky portfolio composition. In our work we provide to the reader a formal description of the stochastic control problem and most importantly we derive an analytical solution assuming that the risky asset dynamics underlying the TVS follow a Black and Scholes (BS) model [2] and that the derivative contract is a European-style option.

Although from a theoretical point of view this solution represents a sub-optimal strategy when dealing with generic dynamics for the risky assets, we show that the BS solution provides a strong baseline as price to practitioners when testing it against numerical methods with different levels of sophistication for solving the control problem. More precisely we numerically study the problem in the general case of a local volatility dynamics [5], [9] by adopting a Reinforcement Learning approach, in particular an *ad hoc* vanilla direct policy

one

strategy for
managing

TVS-based
portfolios
(see the se-
cond minor
remark)

Why?

What is
meant?

algorithm and the more sophisticated proximal policy optimization technique developed in [23].

The paper is organized as follows. In Section 2 we introduce more in detail the manager-bank contract and provide the description of the TVS dynamics in presence of valuation adjustments such as the hedging costs. Then, in Section 3 we introduce the structured class of derivative contracts linked to TVSs, where we describe the arising dynamical control problem for pricing those options. Moreover in this section we derive the BS closed solution for European TVOs in two different ways: one applying the Gyöngy Lemma and the other by writing the Hamilton-Jacobi-Bellman equation. In Section 4 we illustrate how we have applied RL to solve the dynamic control problem, giving a description of the algorithm we have built. We conclude the paper with Section 5 where we present the numerical results obtained in this work for the Black and Scholes model and the local volatility one.

A part of this work has been developed during the Master thesis of one of us (SP), where we had the opportunity to collaborate with Marco Bianchetti and Diego Pierluigi Giovannini from Intesa Sanpaolo Milan, that we thank. Moreover we wish to thank the Italian computing centre Cineca, which approved our ISCRA C project and provided us with the high-computing resources of Marconi100 for the numerical simulations of this contribution.

The opinions here expressed are solely those of the authors and do not represent in any way those of their employers.

2 Target Volatility Strategy

As mentioned before, this work aims to enrich the TVS pricing literature by studying the aspect related to the funding costs coming from hedging the risky assets underlying a target volatility portfolio. Thus we consider the following scenario: a bank selling a protection to a portfolio manager who has his capital invested in a TVS. In our case the fund manager has the freedom of changing the relative weights of the risky asset during the life of the TVS; once the allocation strategy is selected then the volatility targeting algorithm rebalances the risky component of the portfolio with the risk-free one in order to keep the overall portfolio volatility close to a target value. Clients investing in the fund pay a running fee for the service of the fund manager and their capital is protected.

The fund manager usually buys from a bank an option on the TVS to ensure capital protection. For instance, the capital can be protected by buying a put option. In this case, we can write the net asset value (NAV) A_t of the strategy as given by¹

$$A_t := \max\{V_t, K\} = V_t + (K - V_t)^+, \quad (1)$$

¹ Here we neglect discounting factors.

Is this assumption valid?

Put in the
thanks footnote.

where V_t is the price process of the strategy, and K is the guaranteed capital. On the other hand, the fund manager can replicate the payoff by means of the put-call parity by investing the capital in a low-risk asset and buying a call on the strategy

$$A_t = K + (V_t - K)^+ \quad (2)$$

In this way, the TVS is only defined in the two contracts client-fund and fund-bank. The fund manager is not implementing the strategy by trading in the market, and he is not subject to additional costs to access the market. On the other way, the bank is paying such costs since she is actively hedging the call option sold to the manager. The bank trading activity implemented to actively hedge the option requires funding the collateral procedures of the hedging instruments along with any lending/borrowing fee. We remark that the choices of the manager trading activity are stochastic processes since they will depend on the market evolution. Thus, neither the manager strategy nor the bank financing costs for hedging the option can be written in the fund-bank contract. For this reason, the price of a financial product sold by the bank is adjusted to include any valuation adjustment due to the trading activity.

Our aim is to find the most expensive investing strategy from the point of view of financing costs for the bank that the manager could choose in the market. In other words, we want to determine the worst-case scenario for the bank. In the current section, we proceed by defining the price process of the TVS so that we can highlight the impact of valuation adjustments.

2.1 The strategy Price Process

We work on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$ satisfying the usual assumptions for a market model, where \mathbb{P} is the physical probability measure representing the actual distribution of supply and demand shocks on equities prices.

We consider a fund trading a basket of n risky securities with price process $S_t = [S_t^i, i = 1, \dots, n]$ funded with a cash account B_t accruing at r_t . Any dividend paid by the securities is re-invested in the fund. Here, we assume that the TVS is implemented in continuous time, even if in the practice we can implement the strategy only on a discrete set of dates. We introduce the deflated gain process \bar{G}_t^i associated with the risky securities as given by

$$\bar{G}_t^i := \bar{S}_t^i + \bar{D}_t^i, \quad (3)$$

where we define the deflated price and cumulative dividend processes as

$$\bar{S}_t^i := \frac{S_t^i}{B_t}, \quad \bar{D}_t^i := \int_0^t \frac{d\pi_u^i}{B_u} + \int_0^t \frac{d\psi_u^i}{B_u}, \quad (4)$$

where π_t^i represents the cumulative contractual-coupon process paid by the security, and ψ_t^i represents the cumulative valuation adjustments. Since fund managers allocating TVS usually rely on equity assets, here we use the results of [10]

*Make it
shorter*

*What is
meant?*

that analyzes the valuation adjustments for equity products. We can write

$$\psi_t^i := \int_0^t S_u^i \mu_u^i du , \quad (5)$$

where we call μ_u^i cost of carry, which basically represents the hedging costs for the i -th security.

Then, we introduce the strategy price process V_t , and we define the deflated gain process \bar{G}_t^V as given by

$$\bar{G}_t^V := \frac{V_t}{B_t} + \int_0^t \frac{V_u \phi_u}{B_u} du , \quad (6)$$

where ϕ_t are the running fees earned by the fund manager for his activity. We assume that the strategy is self-financing, so that we can write

$$d\bar{G}_t^V = q_t \cdot d\bar{G}_t , \quad (7)$$

where q_t^i is the quantity invested in the i -th security.²

Now, in order to prevent arbitrages, we assume the existence of a risk-neutral measure \mathbb{Q} equivalent to \mathbb{P} under which the deflated gain processes of all traded securities are martingales. Under this assumption, we are able to derive the drift conditions on the security price processes, and in turn on the strategy price process, *that is*

$$\forall u > t \quad \bar{G}_t^i = \mathbb{E}_t [\bar{G}_u^i] \implies dS_t^i = r_t S_t^i dt - d\pi_t^i - d\psi_t^i + dM_t^i , \quad (8)$$

where M_t^i are martingale under \mathbb{Q} . If we substitute this expression for the security dynamics into the definition of the strategy we can check that the price process of the strategy is accruing at a cash account rate r_t compensated for the fund manager fees

$$dV_t = V_t (r_t - \phi_t) dt + dM_t^V , \quad (9)$$

with M_t^V martingale under \mathbb{Q} . Notice that, as expected from non-arbitrage considerations, the coupons paid by each security appear only in the drift of the security price process, but they do not impact the drift of the strategy.

Yet, the strategy priced by V_t cannot be described in the contract between the parties, since Equation (7) depends via the security gain processes on the valuation adjustment ψ_t^i , which is specific to the investor pricing the strategy. Thus, the TVS defined in the contract will be

$$d\bar{I}_t := q_t \cdot \left(d\bar{S}_t + \frac{d\pi_t}{B_t} \right) - \bar{I}_t \phi_t dt \quad \text{with} \quad I_0 = V_0 , \quad (10)$$

leading to the following price process dynamics

$$dI_t = I_t (r_t - \phi_t) dt - q_t \cdot d\psi_t + dM_t^I , \quad (11)$$

² In all formulae we use dot notation for scalar product between vectors, i.e. $a \cdot b = \sum_i a_i b_i$, or between matrix and vector, i.e. $A \cdot b = \sum_j a_{ij} b_j$ or $b \cdot A = \sum_i b_i a_{ij}$.

Schoeller
Möller

with M_t^I martingale under \mathbb{Q} . In this case we observe that I_t depends explicitly both on the valuation adjustments and on the allocation strategy. Indeed, if we substitute the valuation adjustments with their explicit expression (5), we get

$$dI_t = I_t(r_t - \phi_t)dt - q_t \cdot S_t \mu_t dt + dM_t^I, \quad (12)$$

where we can see the dependency on the cost of carry μ_t and on the allocation strategy.

2.2 The Volatility Targeting Constraint

In a typical TVS, the fund manager selects a risky-asset portfolio with a specific time-dependent allocation strategy expressed by means of the vector of relative weights α_t , along with a risk-free asset, which we can identify with the bank account B_t . In the following, for sake of simplicity in exposition, we consider only total-return securities, namely we set $\pi_t = 0$. Thus we can write Equation (10) as given by

$$\frac{dI_t}{I_t} = \omega_t \alpha_t \cdot \frac{dS_t}{S_t} + (1 - \omega_t \alpha_t \cdot \mathbf{1}) \frac{dB_t}{B_t} - \phi_t dt, \quad (13)$$

where $\mathbf{1}$ is a n -dimensional vector of ones and $\omega_t \in [0, 1]$ is determined so that the strategy log-normal volatility is kept constant, namely

$$\omega_t : \text{Var}_t[dI_t] = \bar{\sigma}^2 I_t^2 dt, \quad (14)$$

where $\bar{\sigma}$ is the target volatility value. In practice, this means that the fund manager will select a risky-portfolio choosing α_t equities from the universe where he can trade and after that, his choices will be scaled by the automatic target volatility algorithm ω_t .³

To derive the expression for ω_t we need to assume a generic continuous semi-martingales dynamics under the risk-neutral measure for the underlying securities, so that we can write Equation (8) as

$$\frac{dS_t^i}{S_t^i} = (r_t - \mu_t^i) dt + \nu_t^i \cdot dW_t, \quad (15)$$

where ν_t is an adapted matrix process ensuring the existence of a solution for the stochastic differential equation (SDE) and W_t is a n -dimensional vector of Brownian motions under \mathbb{Q} . Under these assumptions we can derive an expression for ω_t , and we get⁴

$$\omega_t = \frac{\bar{\sigma}}{\|\alpha_t \cdot \nu_t\|}. \quad (16)$$

Hence, putting this last result in the dynamics of I_t we obtain

³ We recall that the universe of assets where the manager can trade and the value of $\bar{\sigma}$ are written in the contract.

⁴ In all formulae the norm for a vector a is defined as $\|a\| := \sqrt{a \cdot a}$.

In this acronym used in the remainder of the manuscript?

for the
derivation
of the
expression for ω_t ensuring
the existence of a solution for
the stochastic differential equation (SDE)
and W_t is a n -dimensional vector of
Brownian motions under \mathbb{Q} . Under these assumptions we can derive an expression
for ω_t , and we get⁴

Motivation

$$\frac{dI_t}{I_t} = \left(r_t - \phi_t - \frac{\bar{\sigma}\alpha_t}{\|\alpha_t \cdot \nu_t\|} \cdot \mu_t \right) dt + \frac{\bar{\sigma}\alpha_t}{\|\alpha_t \cdot \nu_t\|} \cdot \nu_t \cdot dW_t, \quad (17)$$

where we can see, as expected, that the strategy grows at the risk-free rate but for adjustments due to valuation adjustments and fees.

We highlight that, to derive the dynamics expressed in Equation (17), we have not made any assumptions on the risky allocation strategy; thus all this argument is valid for any constraints on the process α_t .

→ or, for instance,
[Provide some example]

3 Derivative Pricing

In this section, we analyze derivative contracts linked to the TVS described previously. In particular, we will focus on European style options, and we will show that under appropriate assumptions it is possible to find a closed form solution for the optimal allocation strategy which maximizes the contract price.

In a general framework, a derivative contract on the TVS with maturity T can be defined as

$$V_0 := \sup_{\alpha} \mathbb{E}_0 \left[\int_0^T D(0, u; \zeta_u) d\pi_u(\alpha) \right], \quad (18)$$

where $D(0, T; \zeta_t)$ is the discount factor with rate ζ_t , inclusive of the derivative valuation adjustments, and π_t is the cumulative coupon process paid by the derivative, and it depends on the allocation strategy since in turn the TVS depends on it via the valuation adjustments. We take the supremum over the strategies since we do not have any information on the future activity of the fund manager.

→ Too synthetic.

3.1 European Options

If the derivative contract depends only on the marginal distribution of I_T at maturity (a European payoff), we are able to calculate an optimal strategy. We consider the following pricing problem

$$V_0 := \sup_{\alpha} \mathbb{E}_0 [D(0, T; \zeta) \Phi(I_T(\alpha))], \quad (19)$$

where Φ is the payoff function of the derivative.

We start by introducing the Markovian projection of the dynamics followed by I_t . We name it I_t^{MP} , and we get by applying the Gyöngy Lemma [12]

$$\frac{dI_t^{\text{MP}}}{I_t^{\text{MP}}} := (r_t - \ell_{\alpha}(t, I_t^{\text{MP}})) dt + \bar{\sigma} dW_t^{\text{MP}} \quad \text{with} \quad I_0^{\text{MP}} = I_0, \quad (20)$$

where the local drift is defined as

$$\ell_{\alpha}(t, K) := \bar{\sigma} \mathbb{E}_0 \left[\frac{\mu_t \cdot \alpha_t}{\|\alpha_t \cdot \nu_t\|} \middle| I_t = K \right], \quad (21)$$

i.e.)

and W_t^{MP} is a Brownian motion under the risk-neutral measure \mathbb{Q} . Notice that the diffusion coefficient collapses to the target volatility value $\bar{\sigma}$. Since European payoffs depend only on the marginal distribution at maturity, they can be calculated only by means of the Markovian projection I_t^{MP} , namely

$$V_0 := \sup_{\alpha} \mathbb{E}_0 [D(0, T; \zeta) \Phi(I_T^{\text{MP}}(\alpha))] . \quad (22)$$

Hence, we have our first result valid only if valuation adjustments can be neglected:

Proposition 1. *A European payoff on the TVS can be calculated by assuming any allocation in the underlying risky basket if all the underlying securities grow under the risk-neutral measure at the risk-free rate without any valuation adjustment, namely if we can write $\mu_t = 0$.*

3.2 Stochastic Optimal Control Problem

In presence of valuation adjustments, we need to solve the full optimization problem. We discretize the optimal strategy α_t as⁵

$$\alpha_t := \sum_k \mathbf{1}_{\{t \in [T_{k-1}, T_k)\}} \alpha_{T_{k-1}} , \quad (23)$$

according to a time grid $\mathcal{T} := \{T_0, \dots, T_k, \dots, T_m\}$ with $T_0 := t$ the pricing date and $T_m := T$ the maturity of the option. Therefore we can apply the dynamic programming principle to express the optimal α_t at time T_{k-1} as

$$\alpha_{T_{k-1}} := \arg \max_{\alpha} \left\{ \mathbb{E}_{T_{k-1}} [D(T_{k-1}, T_k) V_{T_k}(X_{T_k}, I_{T_k}(\alpha)) | X_{T_{k-1}}, I_{T_{k-1}}] \right\} , \quad (24)$$

where V_{T_k} is the option value at time T_k and X is any Markovian state such that the drift and the diffusion coefficient of I_t are a function of (X_t, I_t, α_t) . We calculate $I_{T_k}(\alpha_{T_{k-1}})$ for any given strategy $\alpha_{T_{k-1}}$ by a suitable discretization of (17) starting from $X_{T_{k-1}}$ and $I_{T_{k-1}}$.

Thus the derivative price is given by:

$$V_{T_{k-1}}(X_{T_{k-1}}, I_{T_{k-1}}) = \mathbb{E}_{T_{k-1}} [D(T_{k-1}, T_k) V_{T_k}(X_{T_k}, I_{T_k}(\alpha_{T_{k-1}})) | X_{T_{k-1}}, I_{T_{k-1}}] , \quad (25)$$

while the iteration starts from maturity date where the boundary condition is set equal to the payoff function:

$$V_{T_m} = \Phi(I_{T_m}) . \quad (26)$$

⁵ We use the symbol $\mathbf{1}_A$ for the indicator function of a subset A .

Markov &
risk
neutral.

3.3 Black and Scholes Model

In the time-dependent Black and Scholes model with deterministic rates, we can work with empty X_t , since in this case the portfolio dynamics (17) is Markovian, leading to an optimal strategy α_t^* which depends in principle only on I_t . As a consequence, the local drift defined in Equation (21) can be written as

$$\ell_\alpha(t, K) = \bar{\sigma} \frac{\mu(t) \cdot \alpha(t, K)}{\|\alpha(t, K) \cdot \nu(t)\|}, \quad (27)$$

so that the optimization problem can be solved by looking only at the Markovian projection without simulating all the Brownian motions W_t . Notice that we are indicating the dependency on time in parenthesis to highlight that in this formula all the quantities are deterministic functions of time.

A direct consequence is the following proposition, which is relevant for plain vanilla options on TVS.

Proposition 2. *When the underlying securities follow a Black and Scholes model with deterministic rates, the optimal strategy for a non-decreasing European payoff consists in minimizing the local drift function, independently of the current state I_t*

$$\alpha^*(t) := \arg \min_{\alpha} \frac{\alpha \cdot \mu(t)}{\|\alpha \cdot \nu(t)\|}. \quad (28)$$

Analogously, the optimal strategy for a non-increasing European payoff consists in maximizing the local drift function:

$$\alpha^*(t) := \arg \max_{\alpha} \frac{\alpha \cdot \mu(t)}{\|\alpha \cdot \nu(t)\|}. \quad (29)$$

The absence of stochastic elements in Equation (28) makes the optimal strategy known *a priori* with no simulation needed; in fact one can solve the optimization problem once for all $\forall t \in \mathcal{T}$ just looking at the market data $\mu(t)$ and $\nu(t)$ for the securities. Once α^* is known, then one can price the payoff by the following BS formula

$$V_0^{\text{BS}} = BS(F^{\text{TVS}}(0, T; \alpha^*), K, T, \bar{\sigma}, D(0, T; \zeta)), \quad (30)$$

where $F^{\text{TVS}}(t, T; \alpha)$ is the TVS forward curve defined by

$$F^{\text{TVS}}(t, T; \alpha) = I_t \exp \left[\int_t^T (\bar{r}(u) - \ell_\alpha(u)) du \right], \quad (31)$$

while $BS(F, K, T, \sigma, D)$ is the standard BS formula for a European option with forward curve F , strike K , time to maturity T , volatility σ and discount factor D .

In Figure 1 we provide a comparison among the option price obtained with the optimal strategy (BS^*) of Equation (28) and those with other intuitive strategies (S_A, S_B, S_C). Here we consider the case of an at-the-money call option with spot $I_0 = 1$ EUR, maturity $T = 5$ yr, target volatility $\bar{\sigma} = 5\%$, and a nonnegative constraint on α . The intuitive strategies are

To right
make it
decreas-

to below

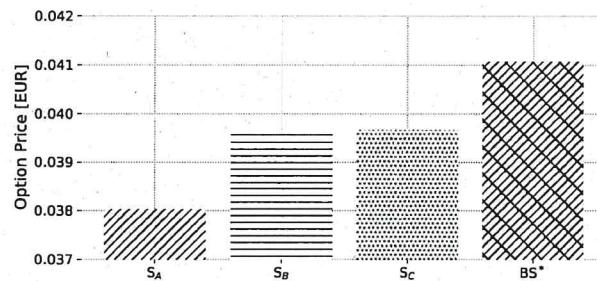


Fig. 1. Comparison of plain vanilla prices on the TVS by adopting different allocation strategies: the optimal Black and Scholes solution (BS^*) of Equation (28) and other intuitive strategies (S_A, S_B, S_C)

- S_A : invest all in the asset with the maximum forward curve at maturity;
- S_B : for each market pillar before maturity invest all in the asset with minimum $\mu(t)$;
- S_C : for each market pillar before maturity invest all in the asset with minimum $\mu(t)/\|\nu(t)\|$.

As the reader can observe, our strategy outperforms any other intuitive one a practitioner might adopt.

Unconstrained Allocation Strategy: Closed Form Solution In absence of constraints on the allocation strategy, we are able to derive a closed form solution to the BS problem (28).

Lemma 1. Let $\mu, \alpha \in \mathbb{R}^n$, $\nu \in \mathbb{R}^{n \times n}$ be a full rank matrix and $\Sigma := \nu\nu^\top$. Then the closed solution of the optimization problem (28) is

$$\alpha^* = -\frac{\Sigma^{-1} \cdot \mu}{\|(\Sigma^{-1} \cdot \mu) \cdot \nu\|}. \quad (32)$$

Proof. Since the argument of the minimum (28) is zero-homogeneous, then we can rewrite the problem as

$$\begin{aligned} & \text{minimize } \alpha \cdot \mu \\ & \text{subject to } \|\alpha \cdot \nu\|^2 = 1. \end{aligned} \quad (33)$$

By setting the Lagrangian function associated with the problem

$$\mathcal{L}(\alpha, \lambda) = \alpha \cdot \mu - \lambda (\|\alpha \cdot \nu\|^2 - 1), \quad (34)$$

Handwritten notes on the left margin:

- "Invest in demand" with an arrow pointing to the first point of the list.
- "Invest some diversification" with an arrow pointing to the second point of the list.
- "Invest whole portfolio next" with an arrow pointing to the third point of the list.

we obtain the first order conditions

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \alpha} = \mu - 2\lambda \Sigma \cdot \alpha = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} = \|\alpha \cdot \nu\|^2 - 1 = 0, \end{cases} \quad (35)$$

Then, by applying simple algebra, we obtain the analytical form for the free optimal strategy

$$\alpha^* = \pm \frac{\Sigma^{-1} \cdot \mu}{\|(\Sigma^{-1} \cdot \mu) \cdot \nu\|}. \quad (36)$$

We take the minus sign to get the minimum value of the TVS local drift while the plus sign for the maximum one (put payoff). \square

Active Asset (or Bang Bang) Solution A closed form solution to the minimization of the local drift correction (28) can also be derived in the common case that all costs of carry are nonnegative and the only constraint on portfolio weights is nonnegativity, which would mean a long-only strategy by the fund manager.

Lemma 2. Let $\mu \in \mathbb{R}^n$ be a vector with nonnegative components, $\nu \in \mathbb{R}^{n \times n}$ be a full rank matrix, and $\Sigma = \nu \nu^\top$. Then

$$\inf_{\alpha \in \mathbb{R}_+^n \setminus \{0\}} \frac{\alpha \cdot \mu}{\|\alpha \cdot \nu\|} = \min_{i \leq n} \frac{\mu_i}{\sqrt{\Sigma_{ii}}}; \quad (37)$$

if \bar{i} is the index which realizes the min, then the infimum is realized by a vector concentrated on the \bar{i} component: $\alpha_i = \delta_{i\bar{i}}$.

Proof. Let us first consider the case in which $\mu = \mathbf{1}$. Since the argument of the infimum is zero-homogeneous, normalizing by $\alpha \cdot \mathbf{1} > 0$ we can restrict to the affine hyperspace $\{\alpha \cdot \mathbf{1} = 1\}$, where the minimization (37) reduces to the maximization of its denominator: the required infimum will be the square root of the reciprocal of

$$\sup \{\|\alpha \cdot \nu\|^2 \mid \alpha \in \mathbb{R}_+^n, \alpha \cdot \mathbf{1} = 1\}. \quad (38)$$

Now we can note that Σ is positive definite, hence $\Sigma_{ij} < \sqrt{\Sigma_{ii}\Sigma_{jj}} \leq \Sigma_{ii}$, which implies

$$\|\alpha \cdot \nu\|^2 = \sum_{i,j=1}^n \alpha_i \alpha_j \Sigma_{ij} \leq \sum_{i,j=1}^n \alpha_i \alpha_j \Sigma_{ii} = \Sigma_{ii}, \quad (39)$$

because $\sum_i \alpha_i = 1$. Since we trivially have equality for $\alpha_i = \delta_{i\bar{i}}$, this concludes the proof of the case $\mu = \mathbf{1}$.

Next, let us consider the case in which all components of μ are strictly positive, and define M as the diagonal matrix with diagonal μ . Then we can rewrite the infimum as a function of $\beta = M\alpha$:

$$\inf_{\beta \in \mathbb{R}_+^n \setminus \{0\}} \frac{\beta \cdot \mathbf{1}}{\|\beta \cdot M^{-1}\nu\|}, \quad (40)$$

What is the financial meaning? Why is this assumption necessary?

which by the first part of the proof equals

$$\min_{i \leq n} \frac{1}{\sqrt{\tilde{\Sigma}_{ii}}} = \min_{i \leq n} \frac{\mu_i}{\sqrt{\Sigma_{ii}}}, \quad \tilde{\Sigma} := M^{-1} \nu \nu^T M^{-1} = M^{-1} \Sigma M^{-1}. \quad (41)$$

Finally, let us consider the general case in which μ may have some components equal to zero. For an arbitrary $\epsilon \geq 0$ let us define

$$f_\epsilon(\alpha) = \frac{\alpha \cdot (\mu + \epsilon)}{\|\alpha \cdot \nu\|}. \quad (42)$$

One can easily note that as $\epsilon \rightarrow 0$, f_ϵ tends to f_0 uniformly on the compact set $\{\alpha \in \mathbb{R}_+^n \mid \alpha \cdot 1 = 1\}$, so that the minimum converges to the minimum on that set. Since we know by homogeneity that the minimum on $\{\alpha \in \mathbb{R}_+^n \mid \alpha \cdot 1 = 1\}$ equals the minimum on $\mathbb{R}_+^n \setminus \{0\}$, we conclude

$$\inf_{\alpha \in \mathbb{R}_+^n \setminus \{0\}} f_0(\alpha) = \lim_{\epsilon \rightarrow 0+} \inf_{\alpha \in \mathbb{R}_+^n \setminus \{0\}} f_\epsilon(\alpha) = \lim_{\epsilon \rightarrow 0+} \min_{i \leq n} \frac{\mu_i + \epsilon}{\sqrt{\Sigma_{ii}}} = \min_{i \leq n} \frac{\mu_i}{\sqrt{\Sigma_{ii}}}. \quad (43)$$

□

3.4 Hamilton-Jacobi-Bellman Equation for Target Volatility Options

In this section, we want to provide to the reader a formal description of the dynamic problem associated with options on target volatility strategies by writing the Hamilton-Jacobi-Bellman (HJB) equation for the derivative price. We prove that from this equation we can recover the same closed formula (28) derived above from the Gyöngy Lemma for the time-dependent BS model.

In full generality, we assume that the time evolution of the Markovian factors governing the problem dynamics is given by a stochastic multidimensional process in \mathbb{R}^n , X_t , which is the unique strong solution to the following Itô SDE

$$dX_t = M(X_t)dt + \Sigma(X_t) \cdot dW_t, \quad (44)$$

where W_t is a n -dimensional Wiener process with independent components. We point out to the reader that with this notation we are including general dynamics models like those with stochastic drift $M(X_t)$ and stochastic diffusive $\Sigma(X_t)$ coefficients.

Let be the dynamics of the securities S_t a generic function of the Markovian factors, namely

$$S_t := f(X_t). \quad (45)$$

In this framework, the TVS price process dynamics is given by the SDE

$$\frac{dI_t}{I_t} = \left(r(X_t) - \frac{\bar{\sigma} \alpha_t}{\|\alpha_t \cdot \nu(X_t)\|} \cdot \mu(X_t) \right) dt + \frac{\bar{\sigma} \alpha_t}{\|\alpha_t \cdot \nu(X_t)\|} \cdot \nu(X_t) \cdot dW_t, \quad (46)$$

where the expression of $\mu(X_t)$ and $\nu(X_t)$ can be recovered by applying the Itô formula to Equation (45).

Given $X := X_t$ and $I := I_t$, we can write the HJB equation for $V := V(t, I, X)$ as follows

$$\frac{\partial V}{\partial t} + \max_{\alpha} \left\{ \left(r(X) - \bar{\sigma} \frac{\alpha \cdot \mu(X)}{\|\alpha \cdot \nu(X)\|} \right) I \frac{\partial V}{\partial I} + (\nabla_X V) \cdot M(X) + \frac{1}{2} \bar{\sigma}^2 I^2 \frac{\partial^2 V}{\partial I^2} \right. \\ \left. + \frac{1}{2} \text{Tr}(\Sigma(X)^T (H_X V) \Sigma(X)) + (\nabla_{X,I} V) \cdot \Sigma(X) \cdot \left(I \bar{\sigma} \frac{\alpha \cdot \nu(X)}{\|\alpha \cdot \nu(X)\|} \right) \right\} = 0, \quad (47)$$

where $\text{Tr}(A)$ is the trace operator of A , $\nabla_X V$ the gradient of V w.r.t. X , $H_X V$ the Hessian matrix of V w.r.t. X and $\nabla_{X,I} V$ is the vector defined by:

$$\nabla_{X,I} V := \left(\frac{\partial^2 V}{\partial X^1 \partial I}, \dots, \frac{\partial^2 V}{\partial X^n \partial I} \right)^T. \quad (48)$$

We take out from the maximum operator all the elements that do not depend on the risky allocation strategy α

$$\frac{\partial V}{\partial t} + r(X) I \frac{\partial V}{\partial I} + (\nabla_X V) \cdot M(X) + \frac{1}{2} \bar{\sigma}^2 I^2 \frac{\partial^2 V}{\partial I^2} + \frac{1}{2} \text{Tr}(\Sigma(X)^T (H_X V) \Sigma(X)) \\ + \bar{\sigma} I \max_{\alpha} \left\{ - \frac{\partial V}{\partial I} \frac{\alpha \cdot \mu(X)}{\|\alpha \cdot \nu(X)\|} + (\nabla_{X,I} V) \cdot \Sigma(X) \cdot \left(\frac{\alpha \cdot \nu(X)}{\|\alpha \cdot \nu(X)\|} \right) \right\} = 0. \quad (49)$$

Equation (49) is the Hamilton-Jacobi-Bellman equation describing the TVS dynamic problem for a generic dynamics of the risky securities underlying the portfolio.

If we assume a time-dependent BS dynamics for the risky equities (μ_t , r_t and ν_t deterministic), then $V = V(t, I)$ and all the derivatives w.r.t. X are zero. Therefore the reduced HJB equation is

$$\frac{\partial V}{\partial t} + r(t) I \frac{\partial V}{\partial I} + \frac{1}{2} \bar{\sigma}^2 I^2 \frac{\partial^2 V}{\partial I^2} + \bar{\sigma} I \max_{\alpha} \left\{ - \frac{\partial V}{\partial I} \frac{\alpha \cdot \mu(t)}{\|\alpha \cdot \nu(t)\|} \right\} = 0; \quad (50)$$

if the payoff is non-decreasing in I by homogeneity of the SDE we get that V is non-decreasing thus the solution is given by

$$\alpha^*(t) = \arg \min_{\alpha} \frac{\alpha \cdot \mu(t)}{\|\alpha \cdot \nu(t)\|}, \quad (51)$$

which is the same result expressed in Equation (28). On the other hand, if the payoff is non-increasing in I then the solution will be the arg max.

Conversely, if we deal with a dynamic model for which the derivative contract V depends on X then the volatility vessor term, namely the second one, in the max operator of Equation (49) is no longer zero and thus one must solve the entire control problem numerically.

In this work we tackle the non-trivial case of a local volatility (LV) model for the S_t -dynamics, such that $\nu_t = \nu(t, S_t)$. We have chosen the LV model since it is well known in financial literature and among practitioners.

Add appropriate references.

4 Reinforcement Learning

As we have discussed in the previous sections, one must resort to numerical approaches to solve the stochastic control problem related to the TVS in the case of general payoffs or risky securities dynamics. The standard approach could be to use classical techniques based on backwards recursion (24)-(25) such as American Monte Carlo [18]. However, their performances degrade exponentially as the dimension n of the problem increases, making prohibitively costly finding the solution to the problem. In our contribution, we adopt a novel technique which is free from the curse of dimensionality and is gaining popularity in many scientific branches for solving stochastic optimal control problems: Reinforcement Learning (RL) [25].

Reinforcement Learning is a branch of Machine Learning which allows an artificial agent to interact with an environment through actions and observations in order to maximize some notion of cumulative reward. In RL, the agent is not told which actions to take but instead, it must discover by trial and error which are the behaviours yielding the highest reward. This is obtained by updating the agent policy π which is a mapping from the environment states to the set of actions. Thus RL is independent of pre-collected data as opposed to other Machine Learning techniques. Because of its nature, RL has been successful in quantitative finance for solving control problems; among the most important RL applications in this field, we refer to [4] as the pioneers in studying self-taught reinforcement trading problems, while to [13] and [17] for hedging derivatives with RL under market frictions.

In our work we adopt two learning strategies to compare their performances in terms of overall reward: an *ad hoc* direct policy algorithm and the state of the art proximal policy optimization (PPO) developed in [23] and [22].

The first method is a specific algorithm developed by us to fit the problem aim to find the optimal option price; this technique can take place as a direct policy optimization method in the wide taxonomy of Reinforcement Learning algorithms. We will go into details in Section 4.1.

On the other hand, the PPO is a high-level actor-critic algorithm well-suited for continuous control problems. It collects a small batch of experiences interacting with the environment to update its decision-making policy. From those interactions with the environment, PPO is able to compute the expected reward and the value function. We will not provide a complete description of this sophisticated learning strategy; for more details, we refer to the authors' papers. In our work, we adopt the PPO implementation found in OPENAI BASELINES.⁶

In the following sections, we describe the way we have formalized the TVS problem in the Reinforcement Learning framework.

4.1 Direct Policy Approach

We consider an episode τ of length $m + 1$ that takes place on a discrete time-grid of fixing times expressed in year fractions $\mathcal{T} := \{T_0, \dots, T_k, \dots, T_m\}$ with $T_0 := 0$

⁶ <https://github.com/openai/baselines>.

It seems too jargon;
give a definition.

and $T_m := T$ maturity of the option. At a given episodic time T_k the RL agent interacts with the environment: it receives a representation of the environment called state s_k and on the basis of that it selects an action a_k sampling from the current policy π^{θ_j} . Here with θ_j we refer to the set of parameters through which we parameterize the agent policy at the j -th algorithm iteration. In our case of study the agent can choose the composition of the risky asset portfolio, so that the policy is the allocation strategy α introduced in Equation (13):

$$a_k = \alpha_{T_k} \quad \forall T_k \in \mathcal{T}, \quad a_k \in \mathcal{A} \subset \mathbb{R}^n. \quad (52)$$

Since the value function of the problem depends on the Markovian state X_t , the portfolio level I_t and time t , our natural choice for the observation state is the following block

$$s_k := [X_{T_k}, I_{T_k}, T_k] \quad \forall T_k \in \mathcal{T}, \quad s_k \in \mathcal{S} \subset \mathbb{R}^{n+2}. \quad (53)$$

In this way, the state contains all the information needed by the agent to take an optimal action, leading to the maximum plain vanilla TVO price.

In this algorithm, we parameterize the agent policy with a feed forward neural network (FFNN), such that θ_j coincides with the hidden weights, s_k with the inputs, and a_k with the output. In this way, we are dealing with a deterministic policy, where its functional form is given by the neural network. The parameters update is performed as follows: the agent collects a finite batch of experiences interacting with the environment in a set of episodes τ , then the loss function is evaluated as

$$L(\theta_j) = D(0, T; \zeta \hat{\mathbb{E}}[\Phi(I_T(\alpha(\theta_j)))]), \quad (54)$$

where the expectation $\hat{\mathbb{E}}[\dots]$ indicates the empirical average on the batch. Then the parameters are updated by plugging the policy into a stochastic gradient ascent algorithm. The choice of Equation (54) is justified by the fact that, when the algorithm will find the optimal set θ^* , then $L(\theta^*)$ will be a good proxy of the optimal option price.

Once the training phase is ended and the agent has selected the optimal policy, we can run a Monte Carlo (MC) simulation with never seen scenarios to price the optimal target volatility option and test if the algorithm has not overfitted the data.

4.2 Proximal Policy Optimization Approach

As for the direct policy approach, we model the pricing problem considering an episode that takes place on the time-grid $\mathcal{T} = \{T_0, \dots, T_k, \dots, T_m\}$ with $T_0 := 0$ and $T_m := T$. Again we choose as observation state the block defined in Equation (53) and the agent policy coincides with the risky allocation strategy α (52). An important difference from the previous method is that: once the agent has selected the action a_k sampled from the current policy, it receives at the next time T_{k+1} a reward r_{k+1} generated by the environment.

On what basis
is it considered good?

Is the policy deterministic?
Why is an expectation used?

Why are different reward functions used
for the two RL-based approaches?
Is not there a loss of comparability
between the results?

In our work, we have defined two different reward functions with the purpose to analyze which could help the agent to learn more efficiently the optimal policy $\pi^* = \pi^{\theta^*}$. Our first definition is

$$r_{k+1} = \begin{cases} D(0, T; \zeta)(I_T(\alpha(\theta_j)) - K)^+ & \text{if } T_{k+1} = T \\ 0 & \text{otherwise} \end{cases} \quad (55)$$

Therefore, during the whole episode, the agent receives a nil reward except at maturity when the reward coincides with the option intrinsic value. This choice may seem too daring because the agent receives a real feedback of its actions only at the end of the episode, increasing the probability to obtain a slow learning. However, if the agent has learnt π^* , the average cumulative reward per episode will coincide with the optimal TVO price.

The second reward function we have defined is

$$r_{k+1} := \gamma^k [V_{BS}(T_{k+1}) - V_{BS}(T_k)], \quad (56)$$

where $\gamma \in [0, 1]$ is an hyper-parameter of the PPO, while $V_{BS}(T_k)$ is a proxy of the residual option price defined by

$$\begin{aligned} V_{BS}(T_k) &:= BS(F^{TVS}(T_k, T; \alpha_{BS}^*, K), K, T - T_k, \bar{\sigma}, D(T_k, T; \zeta)), \\ V_{BS}(T_0) &= 0, \end{aligned} \quad (57)$$

with α_{BS}^* the BS optimal strategy (28) evaluated at the state s_k . In this form the agent actions are hidden inside the term I_t used to compute the TVS forward curve F^{TVS} defined by Equation (31). In this case, the reward function does not suffer of nil values for $0 < T_k < T$: the RL agent always gets a feedback from the environment for its choices. The hyper-parameter γ plays the role of a discount factor in the sense that, as γ approaches to zero, the RL agent will tend to maximize immediate rewards while neglecting possible larger rewards in the future. If we take the cumulative reward per episode and set the PPO parameters⁷ $\gamma = \lambda = 1$ we obtain

$$R(\tau) = \sum_{k=0}^{m-1} r_{k+1} \underset{\gamma=\lambda=1}{=} \sum_{k=0}^{m-1} [V_{BS}(T_{k+1}) - V_{BS}(T_k)] = V_{BS}(T) = (I_T - K)^+, \quad (58)$$

which is equal to the intrinsic value of the option. This result does not depend on the definition of $V_{BS} \forall T_k < T$, but we conjecture that the closer V_{BS} is to the value function, the easier the agent is in learning.

Thus one can train the agent choosing the optimal value for $\gamma, \lambda \in [0, 1]$, and then run, as test phase, a Monte Carlo simulation with $\gamma = \lambda = 1$ and the optimized θ fixed, where, if the agent has learnt π^* , the average of $R(\tau)$ along different episodes will match the optimal undiscounted price of the derivative contract on the TVS.

⁷ We refer to [22] for a more detailed description for the generalized advantage estimation parameter λ .

In the OPENAI BASELINES implementation of the PPO, the agent policy is parameterized again by a neural network; as for the previous method we have chosen an FFNN. In particular with PPO we deal with a stochastic policy whose functional form is a multivariate diagonal Gaussian distribution where the mean $\mu^\theta(s_k)$ is the output vector of the FFNN and the log-standard deviation $\log \sigma$ is an external parameter

$$\pi^\theta(s_k) \sim \mathcal{N}(\mu^\theta(s_k), e^{\log \sigma}). \quad (59)$$

As one can observe, $\log \sigma$ is state-independent, but it is reduced as the number of the PPO update iterations increases. The idea is that the log-standard deviation will be higher at the beginning of the training phase in order to guarantee a good exploration of the action space while it will be lower at the end to avoid too much noise in the proximity of the optimal policy.

The fact that the PPO implementation exploits a stochastic policy ensures us a better exploration of the action space than with the previous approach. Moreover, the algorithm tries to learn an approximator of the on-policy value function as control variate for the training phase. This approximator is an FFNN with the same architecture as the one for the policy.

5 Numerical Investigations

Here we present the numerical results obtained with our proposed methods. We focus our analysis on a European call option on a TVS with the following product details

$$I_0 = K = 1 \text{ EUR}, \quad T = 2 \text{ yr}, \quad \bar{\sigma} = 5\%.$$

Moreover without loss of generality we consider the case of a completely free allocation strategy α . The extension to the constrained case is easy.

It is our aim to investigate the control problem under non-trivial dynamics like the local volatility model where the volatility of the risky asset is also a function of the state. By looking at the HJB equation (49), we expect that the volatility vedor will play a role in finding the optimal solution, giving rise to a non-trivial strategy.

Although in Section 3.3 we have proved that under the Black and Scholes model the Equation (28) solves the control problem, we want to take advantage of this *a priori* solution as a benchmark to gather evidence on the robustness of our Reinforcement Learning approach and to check if our analytical result is correct. Moreover, we use the BS model as numerical laboratory to perform fine-tuning tests for the RL algorithms hyper-parameters and analyze how they impact the final results and performances.

We recall that in both the algorithms we parameterize the agent policy with an FFNN; thus this is completely characterized by the following hyper-parameters: number of hidden layers, number of neurons per hidden layer, activation function per hidden layer. This is due to the fact that in this Reinforcement Learning problem the number of the input neurons is equal to the state

What is meant?

Make it clearer.

Who is it?

Why?

RL

space dimension (53), while the number of the output ones coincides with the action space dimension (52). It is well known in the literature that neural networks give better performances in the training phase if the input data are well normalized [21], [24]. Thus we choose as state s_k the following normalized block

$$[\log(S_{T_k}/F(0, T_k)), I_{T_k}/I_0, T_k] \quad \forall T_k \in \mathcal{T}, \quad (60)$$

where $F(t, T)$ is the forward curve vector of the risky assets from t to T . In Equation (60) we have chosen as Markovian state X_{T_k} the martingale term of the securities dynamics. In this way we have that in the input block the variables have the same order of magnitude.

5.1 Black and Scholes: Hyper-Parameters Fine Tuning

We use the BS environment as toy model to understand which parameters of the RL algorithm play key roles in the training and testing phase. Our first approach has been the direct policy one since it represents a natural way to tackle the problem: since our goal is to find the allocation strategy that maximizes the option price, we update the FFNN weights following the gradient direction of the loss function defined in Equation (54).

We try to investigate the following hyper-parameters: the FFNN architecture, in particular which feature between the depth and the width of the network is more important, the activation function and the learning rate of the optimizer. Moreover, we compare the performances of two well-known optimizers in Machine Learning literature: Nadam [8] and RMSprop [14]. Since we deal with a free allocation strategy α that can assume negative values, we have chosen among the wide variety of activation functions the tanh and the elu. In this way, we can analyze the performance of a saturating activation function and a non-saturating one. Firstly we have performed a grid search on the learning rate starting value and we have found 10^{-3} a good choice in terms of speed of learning and avoiding over-fitting.

In Figures 2 and 3 we present the learning curves of our fine-tuning tests for RMSProp and Nadam respectively. The three lines of each plot display the learning curves for different FFNN architectures: a one hidden layer network with 50 neurons, a 2 hidden layers with 30 and 20 neurons respectively, and a deeper one with three layers with 20, 15, and 5 neurons. Each learning curve is compared with the optimal option price (grey-dashed line) that we have computed with the closed form solution we have derived for the BS model in Section 3.3. We can see that all the learning curves converge to our expected price, providing a numerical demonstration of our theoretical result. More deeply, from the plots we observe that the RMSprop optimizer outperforms the Nadam. Moreover, the tanh activation function seems to be more preferable than the non-vanishing elu. However, more importantly, we have evidence that a deeper architecture of the neural network outperforms the shallow one. All the learning curves we have presented are the best-in-sample results, in terms of performance, of four runs with different random starting guesses for the hidden weights θ . This procedure is necessary since the objective function is not convex.

Give a description of the role in the FFNN.
 Give a description of the role in the FFNN.
 finding

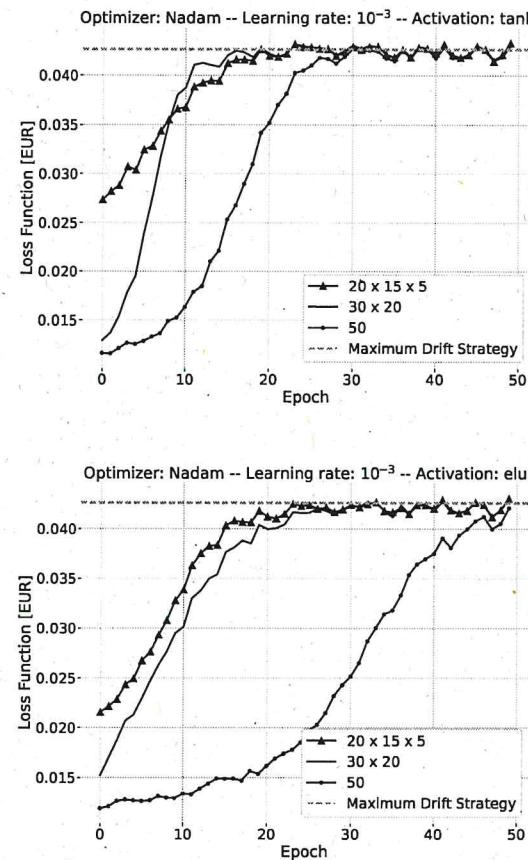


Fig. 2. Learning curves of the direct policy algorithm applied to the Black and Scholes model. The solid lines are the loss functions defined in Equation (54) in function of the number of training epochs. Each continuous line represents a different neural network architecture with tanh activation function (top) and elu activation function (bottom). The optimizer adopted is the Nadam with a learning rate of 10^{-3} . The horizontal grey-dashed line is the conservative option price obtained by maximizing the TVS drift through Equation (36)

We take the $20 \times 15 \times 5$ network with tanh from RMSprop as the best optimized network, and we run a Monte Carlo simulation with 10^6 never-seen scenarios to check if the agent overfits the new data. We report the results in

second
conv
conv

conv

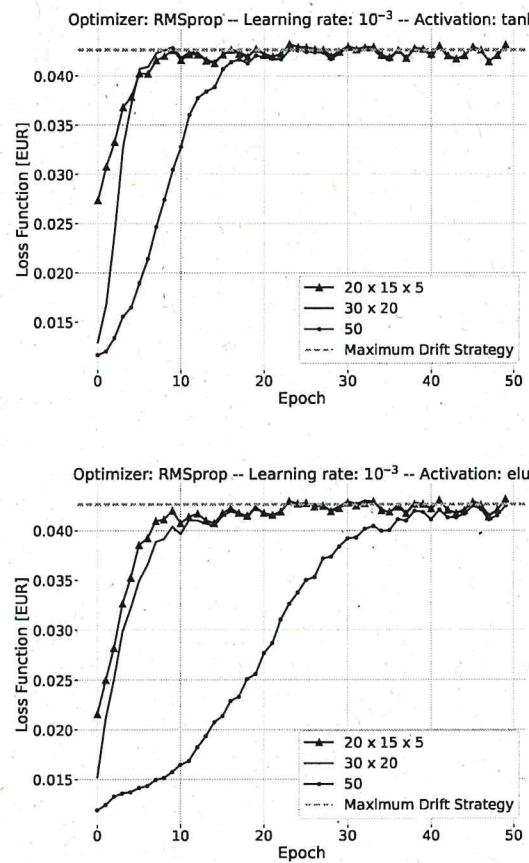


Fig. 3. Learning curves of the direct policy algorithm applied to the Black and Scholes model. The solid lines are the loss values defined in Equation (54) in function of the number of training epochs. Each continuous line represents a different neural network architecture with tanh activation function (top) and elu activation function (bottom). The optimizer adopted is the RMSprop with a learning rate of 10^{-3} . The horizontal grey-dashed line is the conservative option price obtained by maximizing the TVS drift through Equation (36)

Table 1: the RL price is compatible with the closed formula price. Thus the RL agent did not overfit the data during the training phase.

We will take advantage of those fine-tuning results to tackle the local volatility problem.

oncoming
lines

oncoming
lines

Table 1. Comparison of TVO prices under Black and Scholes model: analytical solution price and direct policy reinforcement learning. The option parameters are: $I_0 = K = 1$ [EUR], $T = 2$ [yr] and $\sigma = 5\%$

Pricing method	TVO price [EUR]
Analytical Solution	4.2634×10^{-2}
Direct policy RL	$(4.2624 \pm 0.005) \times 10^{-2}$

5.2 Local Volatility Dynamics

In this section, we study the TVS control problem assuming a local volatility model for the dynamics of the risky assets. Thus in this case we have a diffusive term in Equation (15) that is a deterministic function both of time and state, i.e. the spot price, $v_t = v(t, S_t)$. This additional dependency of volatility makes the problem of finding the optimal strategy non-trivial; in fact if we consider the whole securities smiles then the second-order term in the HJB equation (49) is not zero and thus a closed formula for α^* is not available anymore. Because of that, one must resort to numerical techniques to recover the problem solution. Unlike the BS model where the strategy depends only on time, in LV dynamics there is no unnecessary information provided by the state block in Equation (53) to take the optimal action. Here we consider the same market data (r_t , μ_t and v_t) as in the Black and Scholes environment to study how the optimal solution changes with the dynamics model.

Our first way to tackle the problem is to exploit the results obtained in the BS case. Thus we train by direct policy algorithm a deep $20 \times 10 \times 5$ neural network with tanh activation function and a RMSprop optimizer. We report in Figure 4 the corresponding learning curve (solid line with triangles). From this, we observe that the network has learnt some good policy since the curve grows as the number of training epochs increases until it saturates at a certain value. To try to measure the performance of the policy selected by the agent, we can build a naïf strategy called “baseline”. By looking at the Markovian projection in (20)-(21) and the first term in the HJB (49), a natural choice for the baseline is to maximize the TVS local drift. In other words this coincides by applying the path-wise the Black and Scholes solution of Equation (28); since here we deal with a free allocation strategy, we can simply use our analytical result (36). In the same Figure 4 we report the 99% confidence interval of the MC price obtained with the maximum drift baseline as two red-dashed lines. As we can see, the optimized loss function is compatible with the baseline price. Following the theoretical result of the HJB equation, we can assert that the agent has learnt a sub-optimal policy. Since we LV model differs from the BS one for a corrective term in the HJB (49), we expect that the optimal solution in the LV framework will be in a close region of the maximum drift strategy. Thus we train another agent with the direct policy learning by parameterizing its action with the baseline strategy and choosing a smaller learning rate of 10^{-4} . With this

Make it
deeper

Why is this
constraint necessary?

→ Make
it deeper

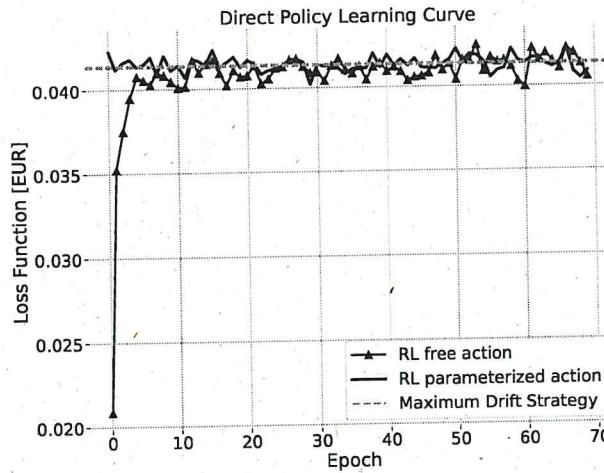


Fig. 4. Learning curves of the direct policy algorithm applied to the local volatility model. The solid line with triangles is the learning curve of an agent with free actions, while the solid one is the learning curve of an agent whose actions are parameterized with the maximum drift strategy. The two agents are parameterized by a $20 \times 10 \times 5$ FFNN with tanh activation function. The loss function is computed according to Equation (54). The optimizer adopted is the RMSprop with a learning rate of 10^{-3} for the solid line with triangles while 10^{-4} for the solid one. The horizontal grey-dashed lines delimit the 99% confidence interval of the MC price obtained by maximizing pathwise the TVS drift through Equation (36)

parameterization, at each observational time $T_k \in \mathcal{T}$ the risky allocation strategy is obtained by summing the network output with the Equation (36). Even in this case, the corresponding learning curve (solid black line in Figure 4) is stuck in the maximum drift strategy. The first possible reason for this behaviour is that the learning strategy of the direct policy approach is too simple for the learning task. The second interpretation is that the volatility vedor in the HJB does not affect significantly the optimum location since it is a second-order term and is lost in the MC error.

Because of that, we change the learning strategy by adopting a more sophisticated one: the PPO algorithm. The big advantage of PPO is that, in addition to the policy, a guess of the value function in Equation (49) is also computed by a parameterization through an FFNN. As for the direct policy approach, we use the BS environment to fine-tune the PPO hyper-parameters. Again we experienced that deeper FFNNs outperform shallow ones and tanh is the most preferable. We report for completeness the values of the other PPO hyper-parameters, for

the description of which we refer to [23]: learning rate 3×10^{-4} , $\lambda = 0.95$, $\epsilon = 0.2$, $c_1 = 0.7$, $c_2 = 0$ and mini-batch size of 2048 episodes.

We have trained a 5 layer FFNN with 8 neurons each with the PPO algorithm in two different environments: one environment generates the reward according to Equation (55), while the other exploits the reward function (56) where we set $\gamma = 0.98$ as discount factor to make the agent prefer immediate rewards. Again, for both environments, we train two different agents: one whose action is given by Equation (52), while the other implements the action parameterization with the baseline. We report in Figure 5 the results of the learning curves. In all the PPO cases, the trained agents seem to be stuck in the sub-optimal maximum drift strategy policy. For the case of the immediate reward function (Figure 5 on the right), we have that the learning curves converge to a saturation value that is not compatible with the baseline price; this is due to the fact that with $\lambda, \gamma \neq 1$ we introduce a bias in the problem. Because of that, we need to run a test MC simulation with freezed agents and $\lambda = \gamma = 1$ to obtain a result to compare with the baseline price. We report the results in Table 2. The Reinforcement Learning agents we have trained with PPO give all prices compatible with the baseline one.

The RL

Table 2. Comparison of TVO prices under local volatility model. The baseline price is obtained by applying path-wise the maximum drift strategy (36). The RL free agent implements the policy defined by Equation (52), while the parameterized agent chooses its actions in terms of the baseline strategy. We label with letter (a) the agents trained in environments with the reward function (55), while with (b) the reward function (56). The option parameters are: $I_0 = K = 1$ [EUR], $T = 2$ [yr] and $\bar{\sigma} = 5\%$

Pricing method	TVO price [EUR]
Baseline	$(4.138 \pm 0.005) \times 10^{-2}$
RL free (a)	$(4.127 \pm 0.005) \times 10^{-2}$
RL parameterized (a)	$(4.131 \pm 0.005) \times 10^{-2}$
RL free (b)	$(4.130 \pm 0.005) \times 10^{-2}$
RL parameterized (b)	$(4.135 \pm 0.005) \times 10^{-2}$

Make it easier

Provide a definition

When γ is close to 1, the option is not short-term, but long-term. Make this point clear.

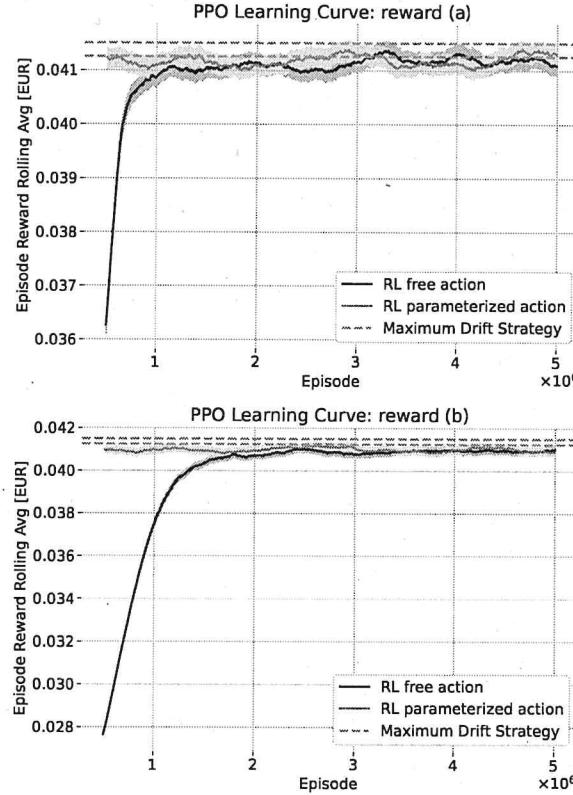


Fig. 5. Learning curves of the PPO algorithm applied to the local volatility model. On the horizontal axis the number of training episodes. The solid lines are the moving average of the realized rewards on the last 10^5 episodes. The shadows represent the 98% confidence intervals. The label *reward (a)* indicates that the RL agent is trained in an environment where the reward function is defined by Equation (55), while *reward (b)* refers to the reward function in Equation (56). The solid black lines are the learning curves of an agent with free actions, while the grey solid ones are the learning curves of an agent whose actions are parameterized with the maximum drift strategy. In both plots, the agents are parameterized by an FFNN with 5 hidden layers with 5 neurons each with tanh activation function. The horizontal grey-dashed lines delimit the 99% confidence interval of the MC price obtained by maximizing path-wise the TVS drift through Equation (36)

is hard to solve because here the control process affects both drift and diffusive coefficients of the controlled process. Despite its complexity, our first contribution is the derivation of a closed form solution of the control problem in a Black and

Scholes framework, which could represent a useful tool for practitioners since it outperforms intuitive trading strategies. We have derived this solution in two different ways: first by applying the Gyöngy Lemma and then by writing the Hamilton-Jacobi-Bellman equation. We numerically studied the problem in the more general local volatility model where the solution is not available and thus a numerical investigation is needed. We tackled the problem by means of the novel Reinforcement Learning techniques, by both the direct policy learning and the proximal policy optimization one. We used the BS model, where the solution is *a priori* known, as benchmark to perform a series of fine-tuning of the RL algorithm hyper-parameters, such as the artificial neural network architecture. We have tested in the LV model the two RL approaches and from our simulations, we have evidence that nor the simple direct policy learning strategy nor the sophisticated PPO are able to outperform our analytical solution applied pathwise. Thus our analytical result for the Black and Scholes model seems to be a good proxy solution also for the local volatility one.

This result seems to be a local optimum from the HJB equation of the problem, since in the LV model the volatility vessor term should influence the RL agent actions. Thus natural development of this work could be to solve the HJB numerically in low dimension in order to check why such sophisticated algorithms are not able to find the global optimum of the problem, or to understand which are the key elements of the problem, such as market data or the payoff function, that can give rise to a solution far from the intuitive one.

References

1. Alberverio, S., Victoria, S., and Wallbaum K.: The volatility target effect in investment-linked products with embedded American-type derivatives. *Investment Management and Financial Innovations* **16-3** (2019) 18-28.
2. Black, F. and Scholes, M.: The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy* **81-3** (1973) 637-654
3. Chew, L.: Target volatility asset allocation strategy. *Society of Actuaries, International News* (2011)
4. Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q.: Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems* **28-3** (2017) 653-664
5. Derman, E. and Kani, I.: Riding on a Smile. *Risk* **7** (1994) 32-39
6. Di Graziano, G., and Torricelli, L.: Target volatility option pricing. *International Journal of Theoretical and Applied Finance* **15-1** (2012) 1-17
7. Di Persio, L., Prezioso, L., and Wallbaum, K.: Closed-end formula for options linked to target volatility strategies. Arxiv preprint (2019)
8. Dozat, T.: Incorporating Nesterov momentum into Adam. *International Conference on Learning Representations* (2016)
9. Dupire, B.: Pricing with a smile. *Risk magazine* **7-1** (1994) 18-20
10. Gabrielli, S., Pallavicini, A., and Scoleri, S.: Funding adjustments in equity linear products. *Risk* (2020)
11. Grasselli, M., and Romo, J.M.: Stochastic skew and target volatility options. *Journal of Futures Markets* **36-2** (2016) 174-193

12. Gyöngy, I.: Mimicking the one-dimensional marginal distributions of processes having an ito differential. *Probability Theory and Related Fields* **4-71** (1986) 501-516
13. Halperin, I.: Q-Learner in the Black-Scholes(-Merton) worlds. *The Journal of Derivatives* **28-1** (2020) 99-122
14. Hinton, G., Srivastava, N., and Swersky, K.: Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on **8-14** (2012)
15. Hocquard, A., Ng, S., and Papageorgiou, N.: A constant volatility framework for managing tail risk. *The Journal of Portfolio Management* **2-39** (2013) 28-40
16. Kim, Y., and Enke, D.: A dynamic target volatility strategy for asset allocation using artificial neural networks. *The Engineering Economist* **4-63** (2018) 273-290
17. Kolm, P.N., and Ritter, G.: Dynamic replication and hedging: a reinforcement learning approach. *The Journal of Financial Data Science* **1-1** (2019) 159-171
18. Longstaff, F., and Schwartz, E.: Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies* **14-1** (2001) 113-147
19. Morrison, S., and Tadrowski, L.: Guarantees and target volatility funds. *Moody's Analytics* (2013)
20. Perchet, R., De Carvalho, R.L., Heckel, T., and Moulin, P.: Predicting the success of volatility targeting strategies: application to equities and other asset classes. *The Journal of Alternative Investments* **3-18** (2016) 21-38
21. Puheim, M., and Madarász, L.: Normalization of inputs and outputs of neural network based robotic arm controller in role of inverse kinematic model. *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics* (2014) 85-89
22. Schulman, J., Moritz, P., Levine, S., Jordan, M.I., and Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. *4th International Conference on Learning Representations* (2016)
23. Schulman, J., Wolski, P., Dhariwal, P., Radford, A., and Klimov, O.: Proximal policy optimization. *Arxiv preprint* (2017)
24. Sola, J., and Sevilla, J.: Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science* **44-3** (1997) 1464-1468
25. Sutton, R.S., and Barto, A.G.: Reinforcement learning: an introduction. MIT Press, Cambridge (2018)
26. Xue, Y.: Target Volatility fund: an effective risk management tool for VA?. *Society of Actuaries, International News* (2012)

Mathematical Methods in Economics and Finance – m²ef

Vol. ..., No. ..., ...

ISSN print edition: 1971-6419 – ISSN online edition: 1971-3878

Web page: <http://www.unive.it/m2ef/> – E-mail: m2ef@unive.it